

TUGAS AKHIR - KI091391

Peramalan Jumlah Kendaraan di Jalan Raya Menggunakan Neural Network Multi Layer Perceptron dengan dan tanpa Regresi Linear

Gregorius Edwadr
NRP 5110 100 188

Dosen Pembimbing
Prof. Dr .Ir. Joko Lianto Buliali, M.Sc.
Diana Purwitasari, S.Kom, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014



FINAL PROJECT - KI091391

FORECASTING NUMBER OF VEHICLE IN ROAD USING MULTILAYER PERCEPTRON NEURAL NETWORK WITH AND WITHOUT LINEAR REGRESSION

**Gregorius Edwadr
NRP 5110 100 188**

**Dosen Pembimbing
Prof. Dr .Ir. Joko Lianto Buliali, M.Sc.
Diana Purwitasari, S.Kom, M.Sc.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014**

LEMBAR PENGESAHAN

Peramalan Jumlah Kendaraan di Jalan Raya Menggunakan Jaringan Saraf Tiruan Multi Layer Perceptron dengan dan tanpa Regresi Linear ITS

TUGAS AKHIR

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada**

**Bidang Studi Komputasi Cerdas dan Visualisasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh:

**Gregorius Edwadr
NRP. 5110 100 188**

Disetujui oleh Pembimbing Tugas Akhir:

1. Prof. Dr .Ir. Joko Lianto Buliah, M.Sc.....
NIP: 19650518 199203 1 003 (Pembimbing 1)
2. Diana Purwitasari, S.Kom, M.Sc.....
NIP: 19810620 200501 1 003 (Pembimbing 2)



**SURABAYA
JULI, 2014**

Peramalan Jumlah Kendaraan di Jalan Raya Menggunakan Neural Network Multi Layer Perceptron dengan dan tanpa Regresi Linear

Nama Mahasiswa : Gregorius Edwadr
NRP : 5110 100 188
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Prof. Dr .Ir. Joko Lianto Buliali,
M.Sc.
Dosen Pembimbing 2 : Diana Purwitasari, S.Kom, M.Sc.

Abstrak

Jumlah kendaraan yang berada kota-kota besar Indonesia semakin lama semakin bertambah tanpa diimbangi oleh pertambahan jumlah jalan yang sebanding. Setelah beberapa tahun masalah ini tidak ditangani munculah masalah kemacetan yang tersebar di banyak kota besar Indonesia. Kemacetan dapat menyebabkan berbagai masalah kepada warga sekitar, oleh karena itu harus ada upaya untuk mengurangi kemacetan.

Hanya ada sedikit cara yang dapat diupayakan untuk mengurangi tingkat kemacetan dengan biaya yang sedikit, misalnya dengan melakukan pengaturan yang baik terhadap nyala lampu lalu lintas. Pengaturan nyala lampu lalu lintas adalah salah satu solusi terbaik yang bisa ditawarkan saat ini namun untuk dapat melakukannya dengan baik dibutuhkan perkiraan jumlah kendaraan yang lewat pada jalan tersebut. Dalam Tugas Akhir ini akan dibahas dua buah model Jaringan Saraf Tiruan yang akan digunakan untuk meramalkan jumlah kendaraan. Metode Jaringan Saraf Tiruan Multi Layer Perceptron akan digunakan untuk meramalkan jumlah kendaraan secara kategorikal dan Jaringan Saraf Tiruan Regresi Linear akan digunakan untuk meramalkan jumlah kendaraan secara numerik.

Uji coba peramalan dilakukan dalam dua jenis yaitu peramalan yang meramalkan jumlah kendaraan secara kategorikal dengan metode Jaringan Saraf Tiruan Multi Layer Perceptron dan peramalan yang meramalkan jumlah kendaraan secara numeric dengan metode Jaringan Saraf Tiruan Regresi Linear. Hasil percobaan untuk jumlah kendaraan yang bersifat kategorikal menunjukan hasil yang cukup baik yaitu dengan tingkat kesalahan 11,7% dan pada peramalan dengan jenis keluaran numerik menghasilkan angka kesalahan Mean Absolute Percentage Error 6,9%.

Kata kunci: Peramalan, Neural Network, Multi Layer Perceptron, Regresi Linear

FORECASTING NUMBER OF VEHICLE IN ROAD USING MULTILAYER PERCEPTRON NEURAL NETWORK WITH AND WITHOUT LINEAR REGRESSION

Student's Name : Gregorius Edwadr
Student's ID : 5110 100 193
Department : Informatics Engineering, FTIF-ITS
First Advisor : Prof. Dr .Ir. Joko Lianto Buliali,
M.Sc.
Second Advisor : Diana Purwitasari, S.Kom, M.Sc.

Abstract

Number of vehicles in many big cities in Indonesia keep increasing everyday but the number of roads does not increase a lot. If this problem could not be solved after several years, which would be a lot of congestion in many cities in Indonesia. As we know congestion can cause variety of problems to local people, therefore there must be an effort to reduce congestion.

There are only few options can be pursued to to reduce the number of congestion which can be done with a little budget, for example by having the traffic lights work properly. Setting the traffic lights is one of the best solutions that can be offered, but to be able to do it, first the approximation of number of vehicles which pass the observed road must be known. This thesis will discuss about two variant methods of Neural Network which will be used to forecast the number of vehicles. Neural Network Multi Layer Perceptron will be used to forecast the number of vehicles in categorical and Neural Network Linear.Regression will be used to forecast the number of vehicles in numerical.

The experiment was conducted in two types of forecasting, the first is the forecasting which predict the number of vehicles categorically by using Neural Network Multi Layer Perceptron and

the second is the forecasting which predict the number of vehicles numerically by using Neural Network Linear Regression. The result for the first forecast showing good accuracy rate, the number of wrong prediction is 11.7% while the second one produces the Mean Absolute Percentage Error 6,9%.

Keywords: Prediction, Neural Network, Multi Layer Perceptron, Linear Regression.

KATA PENGANTAR

Puji syukur kepada Tuhan atas karya keselamatanNya dan karena kasihnya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Peramalan Jumlah Kendaraan di Jalan Raya Menggunakan Jaringan Saraf Tiruan Multi Layer Perceptron dengan dan tanpa Regresi Linear”** dengan baik dan tepat waktu.

Pengerjaan Tugas Akhir ini merupakan suatu pengalaman yang sangat istimewa bagi penulis, karena dalam pengerjaan Tugas Akhir ini penulis bisa melakukan eksplorasi dan implementasi dari hal-hal yang didapatkan selama masa kuliah di Teknik Informatika ITS. terselesaikannya buku Tugas Akhir ini tidak terlepas dari bantuan serta dukungan dari semua pihak. Oleh karena itu, penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Tuhan yang atas karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik dan tepat waktu.
2. Orang tua penulis, yang tidak henti-hentinya memberikan dukungan, semangat, kasih sayang, serta selalu memberikan doa yang dipanjatkan untuk penulis.
3. Bapak Prof. Dr. Ir. Joko Lianto Buliali, M.Sc. selaku dosen pembimbing 1 dan Ibu Diana Purwitasari, S. Kom, M.Sc. selaku dosen pembimbing 2, yang telah memberikan kepercayaan, dukungan, bimbingan, nasehat, serta semangat dikala penulis mengalami kesulitan.
4. Segenap staf dan dosen pengajar di Jurusan Teknik Informatika ITS.
5. Teman-teman penulis, Naufal Aulia Rizal, Tsabbat Aqdami, Nabil, Ibrahim Muhammad Oswaldo, Syahrul Munif, Afrizal, Samodro Bagus, Reza Dwi Putra, Nurul Yuni Arrifa, Gracius Cagar Gunawan yang telah membantu penulis.

6. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya Tugas Akhir ini.

Akhir kata, mohon maaf yang sebesar-besarnya jika terdapat kesalahan pada buku Tugas Akhir ini. Selain itu, penulis juga mengharapkan kritik dan saran yang membangun.

Surabaya, Juli 2014

Gregorius Edwadr

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
Abstrak	ix
Abstract	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR KODE SUMBER.....	xxi
DAFTAR TABEL	xxiii
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat	4
1.5 Metodologi	4
1.6 Sistematika Penyusunan Laporan.....	5
2 BAB II TINJAUAN PUSTAKA	7
2.1 Pengumpulan Data	7
2.1.1 Pembangkitan Data dengan Distribusi Normal	15
2.1.2 Pembangkitan Data dengan Distribusi Eksponensial	15
2.1.3 Pembangkitan Data dengan Distribusi Uniform	16
2.2 Jaringan Saraf Tiruan	17
2.2.1 Lapisan Masukan	17
2.2.2 Laju Pembelajaran.....	19
2.3 Korelasi	19
2.4 Jaringan Saraf Tiruan Multi Layer Perceptron.....	20
2.5 Jaringan Saraf Tiruan Regresi Linear.....	27
3 BAB III PERANCANGAN PERANGKAT LUNAK... 31	31
3.1 Deskripsi Umum Sistem.....	31
3.2 Perancangan Pembangkitan Data	31
3.3 Diagram Alir Jaringan Saraf Tiruan Multi Layer Perceptron	33
3.3.1 Data masukan	33

3.3.2	Inisiasi Bobot.....	33
3.3.3	Pelatihan	36
3.4	Diagram Alir Jaringan Saraf Tiruan Regresi Linear	37
3.4.1	Inisiasi Bobot.....	37
3.4.2	Pelatihan	39
3.5	Perancangan Antar Muka	39
4	BAB IV IMPLEMENTASI.....	45
4.1	Lingkungan Implementasi.....	45
4.2	Lingkungan Implementasi Perangkat Keras.....	45
4.3	Lingkungan Implementasi Perangkat Lunak.....	45
4.4	Implementasi Kode Perangkat Lunak	46
4.4.1	Implementasi Pembangunan Data.....	46
4.4.2	Implementasi Jaringan Saraf Tiruan Multi Layer Perceptron.....	49
4.4.3	Implementasi Jaringan Saraf Tiruan Regresi Linear	56
4.5	Implementasi Antarmuka	60
4.5.1	Antarmuka Pembangunan Data Ujicoba	60
4.5.2	Antarmuka Pemilihan Data Masukan dan Iterasi Maksimum 62	
4.5.3	Antarmuka Jaringan Saraf Tiruan Multi Layer Perceptron.....	63
4.5.4	Antarmuka Jaringan Saraf Tiruan Multi Regresi Linear ...	68
5	BAB V UJI COBA DAN EVALUASI.....	75
5.1	Lingkungan Uji Coba	75
5.2	Data Uji Coba	75
5.3	Skenario Uji Coba	75
5.4	Pengujian dan Evaluasi	77
5.4.1	Lapisan Masukan.....	77
5.4.2	Skenario 1 : Uji Coba Data Masukan Jaringan Saraf Multi Layer Perceptron	78
5.4.3	Skenario 2: Uji Coba Lapisan Tersembunyi Jaringan Saraf Multi Layer Perceptron.....	79
5.4.4	Skenario 3 : Uji Coba Laju Pembelajaran Pada Jaringan Saraf Tiruan Multi Layer Perceptron	80

5.4.5	Skenario 4: Uji Coba Data Masukan Jaringan Saraf Tiruan Regresi Linear	81
5.4.6	Skenario 5 : Uji Coba Laju Pembelajaran Pada Jaringan Saraf Tiruan Regresi Linear	82
5.4.7	Skenario 6 : Uji Coba Pada Data Pelatihan	83
5.5	Analisis.....	83
6	BAB VI KESIMPULAN DAN SARAN.....	85
6.1	Kesimpulan	85
6.2	Saran	85
	DAFTAR PUSTAKA	87
7	LAMPIRAN.....	89
	BIODATA PENULIS	101

DAFTAR TABEL

Table 2.1 Tabel Kolmogorov Smirnov	9
Tabel 2.2 Contoh Jumlah Kendaraan Dengan Distribusi Normal	9
Tabel 2.3 Contoh Jumlah Kendaraan Dengan Distribusi Uniform	14
Tabel 2.4 Contoh pembangkitan dengan distribusi normal	15
Tabel 2.5 Contoh pembangkitan dengan distribusi eksponensial	16
Tabel 3.1 Daftar Simbol dan Keterangannya	32
Tabel 3.2 <i>Pseudocode</i> Pembangkitan Data dengan Distribusi Normal.....	32
Tabel 3.3 <i>Pseudocode</i> Pembangkitan Data dengan Distribusi <i>Uniform</i>	32
Tabel 3.4 <i>Pseudocode</i> Pembangkitan Data dengan Distribusi Eksponensial.....	33
Tabel 4.1 Penyebaran Jumlah Kendaraan di Hari Libur.....	47
Tabel 4.2 Penyebaran Jumlah Kendaraan di Hari Aktif	48
Tabel 5.1 Spesifikasi Lingkungan Pengujian	75
Tabel 5.2 Skenario Uji Coba	76
Tabel 5.3 Kriteria Data Masukan	77
Tabel 5.4 Kombinasi Data Masukan	78
Tabel 5.5 Nilai Kesalahan pada Percobaan Data Masukan Jaringan Saraf Tiruan Multi Layer Perceptron.....	79
Tabel 5.6 Nilai Kesalahan pada Percobaan Jumlah Neuron pada Lapisan Tersembunyi	80
Tabel 5.7 Nilai Kesalahan pada Percobaan Jaringan Saraf Tiruan Multi Layer Perceptron pada Laju Pembelajaran	81
Tabel 5.8 Nilai Kesalahan pada Percobaan Data Masukan Jaringan Saraf Tiruan Regresi Linear	82
Tabel 5.9 Nilai Kesalahan pada Percobaan Jaringan Saraf Tiruan Regresi Linear pada Laju Pembelajaran	82
Tabel 5.10 Nilai Kesalahan pada Percobaan pada Data Uji	83
Tabel L1 Contoh data kategori A (1).....	89
Tabel L2 Contoh data kategori A (2).....	92
Tabel L3 Contoh data kategori A (3).....	94

Tabel L4 Contoh data kategori B (1)	95
Tabel L5 Contoh data kategori B (2)	96
Tabel L6 Contoh data kategori B (3)	97
Tabel L7 Contoh data kategori B (4)	98
Tabel L8 Contoh data kategori B (5)	99
Tabel L9 Detail waktu pengambilan data	99

DAFTAR GAMBAR

Gambar 2.1 Grafik data pada Tabel 2.1.....	11
Gambar 2.2 Grafik penyebaran data kedatangan kendaraan dalam detik.....	13
Gambar 2.3 Struktur Jaringan Saraf Tiruan Multi Layer Perceptron	22
Gambar 2.4 Struktur Jaringan Saraf Tiruan Regresi Linear	28
Gambar 3.1 Diagram Alir Memberi Kriteria Data Masukan.....	34
Gambar 3.2 Diagram Alir inisiasi bobot Jaringan Saraf Tiruan Multi Layer Perceptron.....	35
Gambar 3.3 Diagram Alir Jaringan Saraf Tiruan Multi Layer Perceptron.....	36
Gambar 3.4 Diagram Alir Inisiasi Bobot Neural Netwrok Regresi Linear.....	38
Gambar 3.5 Diagram Alir Jaringan Saraf Tiruan Regresi Linear	40
Gambar 3.6 Tampilan awal pada tab Inisiasi	41
Gambar 3.7 Tampilan awal tab Regresi	43
Gambar 4.1 Tampilan Antarmuka Halaman Pembangkitan Data Uji Coba.....	61
Gambar 4.2 Antarmuka pemilihan data masukan dan iterasi maksimum	62
Gambar 4.3 Bagian inisiasi bobot Jaringan Saraf Tiruan Multi Layer Perceptron	64
Gambar 4.4 Bagian detail inisiasi bobot Jaringan Saraf Tiruan Multi Layer Perceptron.....	65
Gambar 4.5 Tampilan nilai kesalahan Jaringan Saraf Tiruan Multi Layer Perceptron	66
Gambar 4.6 Hasil peramalan pada Jaringan Saraf Tiruan Multi Layer Perceptron	67
Gambar 4.7 Tampilan nilai kesalahan Jaringan Saraf Tiruan Regresi Linear	70
Gambar 4.8 Hasil Peramalan 1 Jaringan Saraf Tiruan Regresi Linear.....	71

Gambar 4.9 Hasil Peramalan 2 Jaringan Saraf Tiruan Regresi Linear.....	72
Gambar 4.10 Grafik bobot Neural Netwok Regresi Linear.....	73

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Kode Untuk Membangkitkan Data dengan Distribusi Normal.....	49
Kode Sumber 4.2 Implementasi Kode Untuk Membangkitkan Data dengan Distribusi Uniform.....	49
Kode Sumber 4.3 Implementasi Kode Untuk Membangkitkan Data dengan Distribusi Eksponensial	49
Kode Sumber 4.4 Implementasi iterasi maksimal	50
Kode Sumber 4.5 Implementasi fungsi makeinput	51
Kode Sumber 4.6 Implementasi fungsi initialweight	51
Kode Sumber 4.7 Implementasi fungsi randoming	52
Kode Sumber 4.8 Implementasi fungsi pelatihan Jaringan Saraf Tiruan Multi Layer Perceptron.....	53
Kode Sumber 4.9 Implementasi #region klasifikasi pada Kode Sumber 4.8.....	54
Kode Sumber 4.10 Implementasi #region hitungErrorOutputDanHidden pada Kode Sumber 4.8	55
Kode Sumber 4.11 Implementasi #region updateBobosDanBias pada Kode Sumber 4.8	56
Kode Sumber 4.12 Implementasi penghitungan korelasi	57
Kode Sumber 4.13 Implementasi normalisasi korelasi	58
Kode Sumber 4.14 Implementasi fungsi pelatihan Jaringan Saraf Tiruan Regresi Linear.....	59
Kode Sumber 4.15 Implementasi #region prediksiDanMenghitungError pada Kode Sumber 4.14	59
Kode Sumber 4.16 Implementasi #region UpdateBobotDanBias pada Kode Sumber 4.14	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemacetan adalah masalah yang dimiliki oleh banyak kota besar yang ada di Indonesia. Kemacetan dapat disebabkan oleh banyak hal antara lain arus yang melewati jalan telah melampaui kapasitas jalan, terjadi banjir sehingga memperlambat kendaraan, ada perbaikan jalan, bagian jalan tertentu yang longsor, karena adanya pemakai jalan yang tidak tahu aturan lalu lintas, adanya parkir liar dari sebuah kegiatan dan pengaturan lampu lalu lintas yang bersifat kaku yang tidak mengikuti tinggi rendahnya arus lalu lintas. Dampak yang ditimbulkan oleh kemacetan sendiri antara lain kerugian waktu karena kecepatan perjalanan yang rendah, pemborosan energi, keausan kendaraan lebih tinggi karena waktu yang lebih lama untuk jarak yang pendek, penggunaan rem yang lebih tinggi, meningkatkan polusi udara karena pada kecepatan rendah konsumsi energi lebih tinggi, meningkatkan stres pengguna jalan, mengganggu kelancaran kendaraan darurat.

Dari penyebab-penyebab kemacetan diatas, ada masalah yang hanya bisa diselesaikan dengan pembangunan secara fisik dan ada juga yang dapat diselesaikan tanpa melakukan pembangunan secara fisik. Misalnya dengan mengatur lama lampu lalu lintas agar bekerja dengan optimal atau dengan menempatkan petugas-petugas lalu lintas di waktu dan jalan yang tepat. Untuk kedua solusi diatas pertama-tama harus diketahui banyak kendaraan yang lewat pada suatu jalan dalam suatu waktu tertentu. Prediksi yang baik tentang jumlah kendaraan yang melewati suatu jalan pada suatu waktu tertentu akan sangat membantu untuk melakukan banyak kegiatan yang berhubungan dengan menurunkan tingkat kemacetan. Salah satu metode untuk menyelesaikan masalah peramalan jumlah kendaraan adalah Jaringan Saraf Tiruan. Untuk melakukan peramalan dengan metode Jaringan Saraf Tiruan akan digunakan data jumlah kendaraan pada hari yang sama di minggu-minggu sebelumnya, data jumlah kendaraan pada beberapa menit

sebelumnya, dan data jumlah kendaraan pada beberapa hari yang lalu. Semua data yang akan digunakan sebagai data masukan akan diberikan bobot yang berbeda-beda berdasarkan nilai korelasi yang telah dihitung sebelumnya.

Beberapa penelitian telah dilakukan sebelumnya, misalnya pada penelitian [1] yang telah dilakukan tahun 2006 di kota Melbourne, Australia, penelitian dilakukan dengan tujuan mengurangi tingkat kemacetan di kota tersebut. Metode yang digunakan adalah *Prioritising and assessing expansion initiatives relating to congestion*. Tujuan dari metode ini adalah mengoptimalkan semua jalan yang ada sehingga dapat mengurangi kemacetan. Kesimpulan yang diperoleh adalah semakin banyak variasi dari rute alternatif yang digunakan maka kemacetan pada jalan-jalan utama dapat berkurang, namun dalam sepuluh tahun terakhir, jumlah kendaraan terus meningkat sehingga dampak penggunaan rute-rute alternatif menjadi kurang signifikan.

Masalah yang sama dialami kota Istanbul, Turki juga sama sehingga dilakukan juga penelitian yang serupa di kota tersebut. Tujuan dari penelitian ini adalah meramalkan jumlah kendaraan sehingga dapat diambil keputusan untuk mengurangi kemacetan ini dimasa yang akan datang. Metode peramalan yang dilakukan adalah Jaringan Saraf Tiruan Multi Layer Perceptron, Jaringan Saraf Tiruan Elman Rekuren dan *Non linear Auto Regressive and eXogenous Input Type ANN*. Hasil terbaik didapatkan dengan menggunakan Jaringan Saraf Tiruan Multi Layer Perceptron dengan metode pembelajaran Levenberg Marquardt. [2]

Pada percobaan lainnya, metode L-shapes net dan Z-shapes net [3] diterapkan untuk mencari probabilitas kemacetan yang akan terjadi. Metode ini memodelkan jumlah kendaraan menjadi bentuk L dan Z kemudian menghitung peluang masing-masing titik yang dilalui oleh kedua model tersebut. Jalan raya dimodelkan menjadi sebuah persegi panjang dengan panjang dan lebar tertentu. Tujuan dari metode adalah untuk melakukan peramalan jumlah kendaraan disuatu jalan raya (yang telah dimodelkan menjadi sebuah persegi panjang) dalam suatu rentang waktu.

Tujuan dari Tugas Akhir ini adalah untuk membuat Jaringan Saraf Tiruan yang dapat meramalkan jumlah kendaraan yang melewati suatu jalan. Setelah dapat diramalkan jumlah kendaraan yang akan melewati jalan-jalan di suatu kota maka akan dapat dilihat apakah pada jalan-jalan tertentu akan mengalami kemacetan atau tidak. Banyak kebijakan yang dapat diterapkan untuk mengurangi kemacetan jika memang telah diketahui sebelumnya bahwa kemacetan akan terjadi. Mengatur lampu lalu lintas dan menempatkan petugas di jalan-jalan yang akan mengalami kemacetan adalah solusi yang baik dengan anggaran yang sedikit.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam menyelesaikan tugas akhir ini adalah:

1. Bagaimana mengumpulkan data jumlah kendaraan dan membangkitkannya?
2. Bagaimana memilih bobot dengan bantuan korelasi dan rentang nilai tertentu?
3. Bagaimana membangun Jaringan Saraf Tiruan Multi Layer Perceptron dan Jaringan Saraf Tiruan Regresi Linear?

1.3 Batasan Masalah

Asumsi dan ruang lingkup permasalahan yang dikerjakan dalam tugas akhir ini adalah:

1. Karena keterbatasan waktu, sebagian data uji coba dibangkitkan secara acak dan sebagian lainnya didapatkan dari pengamatan di sebuah jalan.
2. Faktor penentu dalam peramalan adalah jumlah kendaraan pada waktu-waktu sebelum waktu yang akan diramalkan dan jenis hari libur atau tidak.

1.4 Tujuan dan Manfaat

Tujuan pembuatan Tugas Akhir ini adalah untuk mengimplementasikan metode Jaringan Saraf Tiruan untuk meramalkan jumlah kendaraan pada jalan-jalan yang telah diamati, baik secara kategorikal (dengan menggunakan metode Jaringan Saraf Tiruan Multi Layer Perceptron) dan secara numerik (dengan menggunakan metode Jaringan Saraf Tiruan Regresi Linear)

Manfaat yang diharapkan dapat tercapai dengan selesainya Tugas Akhir ini adalah dapat diambil beberapa keputusan untuk menyelesaikan masalah kemacetan. Beberapa contoh dari keputusan-keputusan yang dimaksud adalah dengan mengoptimalkan lampu merah, menempatkan petugas lalu lintas di jalan-jalan tertentu atau bahkan pengalihan kendaraan ke rute-rute lainnya.

1.5 Metodologi

Ada beberapa tahap dalam proses pengerjaan Tugas Akhir ini. Berikut ini adalah tahap-tahap dalam pembuatan Tugas Akhir.

a. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi dan pembelajaran yang akan digunakan pada Tugas Akhir ini. Studi literatur meliputi diskusi dan pemahaman mengenai topik Tugas Akhir ini, diantaranya mengenai:

- a. Algoritma Jaringan Saraf Tiruan Multi Layer Perceptron
- b. Algoritma Jaringan Saraf Tiruan dengan Regresi Linear

b. Desain Sistem

Tahap ini merupakan perancangan sistem dengan menggunakan studi literatur dan mempelajari konsep aplikasi yang akan dibuat. Dengan bekal teori, metode, dan informasi yang sudah terkumpul pada tahap sebelumnya diharapkan dapat membantu dalam proses perancangan sistem.

c. Implementasi

Implementasi merupakan tahap penerapan bagaimana melakukan peramalan terhadap data jumlah kendaraan dengan metode Jaringan Saraf Tiruan Multi Layer Perceptron dengan dan tanpa Regresi Linear.

d. Uji Coba dan Evaluasi

Tingkat keberhasilan untuk kedua metode yang diimplementasikan adalah berbeda. Pada metode Jaringan Saraf Tiruan Multi Layer Perceptron, tingkat keberhasilan dinilai dari seberapa banyak jumlah kelas yang diramalkan secara benar. Pada metode Jaringan Saraf Tiruan Regresi Linear tingkat keberhasilan dinilai dari seberapa kecil *Mean Absolute Percentage Error* yang dihasilkan.

e. Penyusunan Laporan Tugas Akhir

Tahap ini digunakan untuk membuat laporan Tugas Akhir yang berisi metode, dasar teori, dan hasil yang didapatkan selama pengerjaan Tugas Akhir.

1.6 Sistematika Penyusunan Laporan

Buku Tugas Akhir ini disusun dengan sistematika laporan sebagai berikut:

1. Bab I. Pendahuluan

Bab ini meliputi latar belakang masalah, rumusan permasalahan, batasan masalah, tujuan dan manfaat pembuatan Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan laporan Tugas Akhir.

2. Bab II. Tinjauan Pustaka

Bab ini meliputi dasar teori dan penunjang yang berkaitan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

3. Bab III. Perancangan Perangkat Lunak

Bab ini membahas desain dari sistem yang akan dibuat meliputi diagram alir dan perancangan antarmuka sistem.

4. Bab IV. Implementasi

Bab ini membahas implementasi dari desain sistem yang dilakukan pada tahap desain, meliputi kode dan implementasi antarmuka dari perangkat lunak.

5. Bab V. Uji Coba dan Evaluasi

Bab ini membahas uji coba dari perangkat lunak yang dibuat dengan melihat keluaran yang dihasilkan oleh perangkat lunak, analisis, dan evaluasi untuk mengetahui kemampuan perangkat lunak.

6. Bab VI. Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil uji coba yang dilakukan serta saran untuk pengembangan lebih lanjut perangkat lunak.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan.

2.1 Pengumpulan Data

Sumber data didapat dari mengamati langsung jumlah kendaraan yang melewati beberapa jalan yang telah ditentukan. Dari pengamatan, akan didapat jumlah kendaraan yang melewati jalan yang diamati setiap satu menit selama selang waktu yang telah ditentukan. Pengamatan telah dilakukan diwaktu-waktu tertentu yang dianggap dapat menggambarkan jumlah kendaraan yang melewati jalan tersebut diwaktu sibuk dan tidak.

Dari data yang telah diperoleh, dilakukan pembangkitan bilangan dengan pola penyebaran distribusi yang sesuai dengan data tersebut. Pencarian pola persebaran data dilakukan dengan pemodelan masukan (*input modeling*) [4]. Pembangkitan data akan dilakukan untuk jumlah kendaraan setiap satu menit kemudian dari data yang telah diperoleh dikelompokkan setiap 15 menit untuk kemudian dijadikan data masukan pada proses selanjutnya.

Contoh data dengan distribusi normal

Misalnya selama 20 menit, setiap menitnya jumlah kendaraan yang melewati suatu jalan 12, 20, 11, 10, 18, 12, 14, 12, 10, 11, 13, 12, 11, 10, 7, 12, 12, 14, 16 dan 13. Jika dicurigai kumpulan data tersebut terdistribusi secara normal maka harus dilakukan perhitungan untuk membuktikan hipotesa tersebut. Perhitungan pemodelan masukan untuk data diatas dengan hipotesa bahwa data tersebut tersebar secara normal dapat dilihat pada Tabel 2.1. Sebelum melakukan perhitungan terutama data-data tersebut harus diurutkan dari kecil ke besar.

Kolom pertama dalam Tabel 2.2 adalah frekuensi komulatif yang berisi angka 1 sampai 20 yang mengindikasikan indeks data. Kolom kedua adalah data yang telah diurutkan. Kolom ketiga adalah $F(x_i)$. $F(x_i)$ adalah fungsi normalisasi data untuk merubah data kedalam rentang 0 sampai 1. $F(x_i)$ dapat dihitung dengan Persamaan 2.1.

$$F(x_i) = \frac{(x_i - x_1)}{(x_n - x_1)} \quad (2.1)$$

x_i adalah data ke- i , x_1 adalah data terkecil dan x_n adalah data terbesar, n adalah jumlah data total(dalam kasus ini adalah 20). Dapat dikatakan bahwa $F(x_i)$ adalah proses untuk membuat nilai x_i menjadi sebuah nilai baru dengan rentang 0 sampai 1. Kolom keempat adalah i/n dengan i adalah frekuensi komulatif(kolom 1). Kolom kelima adalah hasil pengurangan dari kolom keempat dikurangi kolom ketiga. Kolom keenam berisi $F(x_i) - (\frac{i-1}{n})$. Hasil yang dapat diperoleh dari hasil perhitungan ini adalah nilai maximum dari kolom kelima dan keenam, yaitu 0,31 dan 0,18. Parameter α adalah konstanta yang ditentukan sendiri, dalam kasus ini α akan bernilai 0.05. Grafik persebaran data dari Tabel 2.2 dapat dilihat pada Gambar 2.1.

Setelah mendapatkan hasil perhitungan diatas maka akan dilihat juga nilai yang berasal dari Tabel Kolmogorov-Smirnov [4] untuk jumlah data 20 dan α 0,05. Nilai pada tabel untuk parameter jumlah data dan α yang ditentukan seperti diatas adalah 0,294 seperti yang ditunjukkan oleh Tabel 2.1. Nilai dari tabel dibandingkan dengan nilai terkecil antara 0,31 dan 0,18 yaitu 0,18. Karena nilai hitung kurang dari nilai dari tabel maka kesimpulannya adalah data tersebut diterima sebagai data yang terdistribusi secara normal.

Contoh lain dari pemodelan data yang terdistribusi secara normal dapat dilihat pada Tabel L4, Tabel L5 dan Tabel L6 pada Lampiran.

Table 2.1 Tabel Kolmogorov Smirnov

n	$\alpha = 0,20$	$\alpha = 0,10$	$\alpha = 0,05$	$\alpha = 0,02$	$\alpha = 0,01$
1	0,900	0,950	0,975	0,990	0,995
5	0,447	0,509	0,563	0,627	0,669
10	0,323	0,369	0,409	0,457	0,486
15	0,266	0,304	0,338	0,377	0,404
20	0,232	0,265	0,294	0,329	0,352
25	0,208	0,238	0,264	0,295	0,317
30	0,190	0,202	0,242	0,270	0,290

Tabel 2.2 Contoh Jumlah Kendaraan Dengan Distribusi Normal

i	x_i	$F(x_i)$	i/n	$\frac{i}{n} - F(x_i)$	$F(x_i) - (\frac{i-1}{n})$
1	7	0	0,05	0,05	0
2	10	0,23	0,1	-0,13	0,18
3	10	0,23	0,15	-0,08	0,13
4	10	0,23	0,2	-0,03	0,08
5	11	0,31	0,25	-0,06	0,11
6	11	0,31	0,3	-0,01	0,06
7	11	0,31	0,35	0,04	0,01
8	12	0,38	0,4	0,02	0,03
9	12	0,38	0,45	0,07	-0,02
10	12	0,38	0,5	0,12	-0,07
11	12	0,38	0,55	0,17	-0,12
12	12	0,38	0,6	0,22	-0,17
13	12	0,38	0,65	0,27	-0,22
14	13	0,46	0,7	0,24	-0,19
15	13	0,46	0,75	0,29	-0,24
16	14	0,54	0,8	0,26	-0,21
17	14	0,54	0,85	0,31	-0,26
18	16	0,69	0,9	0,21	-0,16
19	18	0,85	0,95	0,1	-0,05
20	20	1	1	0	0,05

Cara untuk membangkitkan data yang terdistribusi secara normal dapat menggunakan Persamaan 2.2. U adalah hasil pembangkitkan sebuah angka yang terdistribusi secara normal. Cara untuk mencari nilai U dapat dilihat pada Persamaan 2.3. r_1 dan r_2 adalah dua buah konstanta dengan rentang 0 sampai 1.

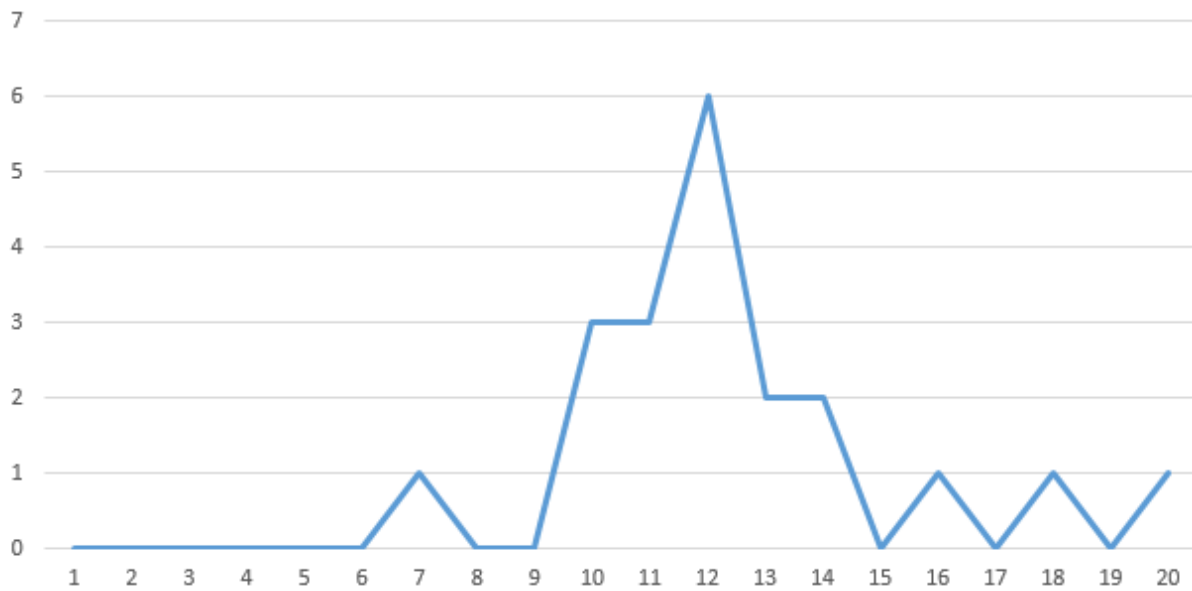
$$x_i = \mu + (\sigma \times U) \quad (2.2)$$

$$U = \sqrt{-2 \times \log r_1} \times \sin(2 \times \pi \times r_2) \quad (2.3)$$

Contoh data dengan distribusi eksponensial

Berikut adalah contoh data yang telah diambil. 1,5,7,8,9,13,15,16,28,29,34,40,43,50,52,55,56,57,63,67,68,69,70,73,78,85,89,90,93,98,103,113,116,118,119,122,126,131,133,136,138,141,145,146,148,150,155,160,161,164,166,167,172,175,181,186,192,198,202,203,215,215,216,223,225,231,233,236,238,242,243,244,245,247,248,253,261,262,263,266,269,274,276,281,285,293,295. Data-data tersebut adalah waktu dimana tercatat ada kendaraan yang lewat, pengamatan diatas dilakukan selama 5 menit dengan tercatat ada 87 kendaraan yang lewat. Grafik dari data diatas dapat dilihat pada Gambar 2.2.

Uji pemodelan input untuk membuktikan bahwa data tersebut terdistribusi secara eksponensial adalah dengan menghitung D^- dan D^+ . D^- dapat dihtung dengan menggunakan Persamaan 2.4 dan D^+ dapat dihitung dengan menggunakan Persamaan 2.5.



Gambar 2.1 Grafik data pada Tabel 2.1.

$$D^- = \sum_{i=1}^n \frac{i}{n} - x_i \quad (2.4)$$

$$D^+ = \sum_{i=1}^n x_i - \frac{i-1}{n} \quad (2.5)$$

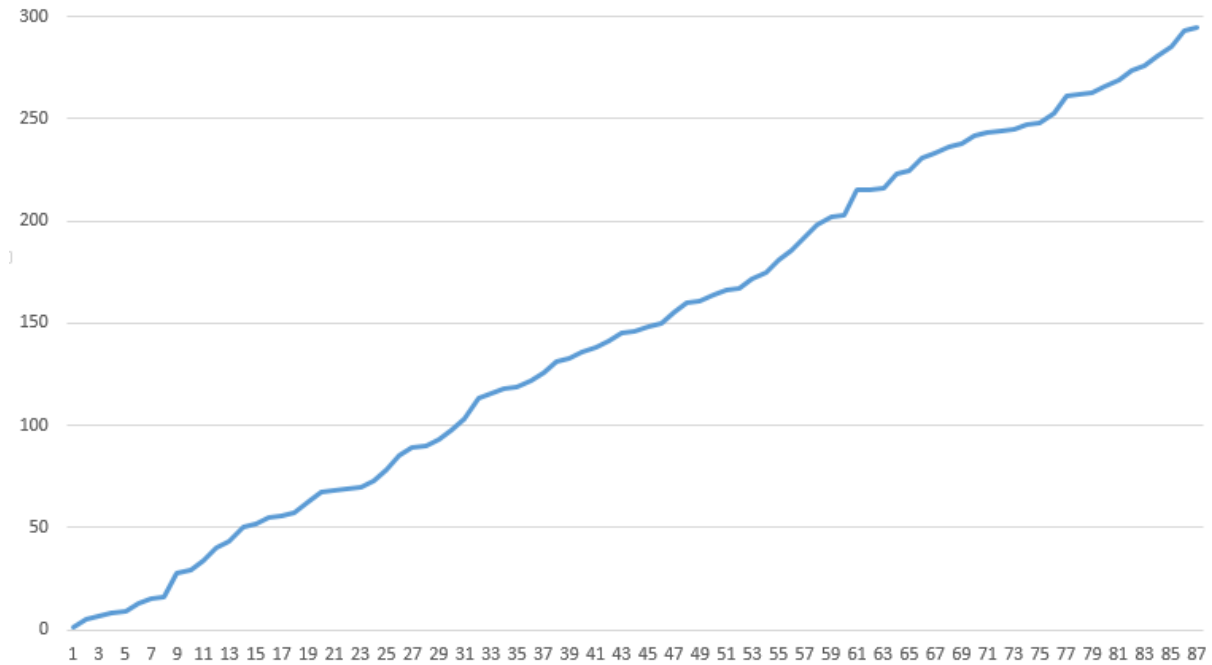
Untuk setiap data tersebut dapat dibuat bentuk x_i dengan membaginya dengan 300 dimana 300 adalah lama waktu pengamatan (5 menit = 300 detik). i adalah indeks yang dimulai dari 1 sampai dengan 87, n adalah jumlah data yaitu 87. Untuk setiap data akan hitung nilai D^- dan D^+ . Dalam kasus ini jumlahan D^- adalah 0,1438621 dan jumlahan D^+ adalah -0,18713. Nilai maksimum dari kedua nilai tersebut adalah 0,1438621 dan akan dibandingkan dengan nilai dari tabel. Karena datanya berjumlah 87 baris dan tingkat signifikansi 95% maka untuk mencari nilai dari Tabel Kolmogorov-Smirnov [4] digunakan Persamaan 2.6 yang disimbolkan dengan X dan menghasilkan nilai 0,145807. Karena nilai hitung lebih kecil dari nilai tabel maka terbukti bahwa data tersebut terdistribusi secara eksponensial. n adalah banyaknya data, dalam kasus ini adalah 87.

$$X = \frac{1,36}{\sqrt{n}} \quad (2.6)$$

Contoh lain dari pemodelan data yang terdistribusi secara eksponensial dapat dilihat pada Tabel L1, Tabel L2 dan Tabel L3 pada Lampiran.

Cara pembangkitan bilangan dengan distribusi eksponensial dapat dilakukan dengan menggunakan Persamaan 2.7. μ adalah rata-rata dari data dan r adalah konstanta dengan rentang 0 sampai 1.

$$x_i = -\log(1-r) \times \mu \quad (2.7)$$



Gambar 2.2 Grafik penyebaran data kedatangan kendaraan dalam detik

Contoh data dengan distribusi uniform

Contoh kasus untuk data yang terdistribusi secara *uniform* dapat dilihat pada Tabel 2.3. $Fa(x_i)$ adalah frekuensi relatif kumulatif yang didapat dengan menjumlahkan semua frekuensi sebelum i . $Fe(x_i)$ adalah Frekuensi Teoritis yang didapat dengan membagi x_i dengan sebuah nilai yang dianggap paling maksimum, dalam kasus ini adalah 21. Untuk membuktikan bahwa kumpulan data tersebut terdistribusi secara *uniform* maka nilai yang terbesar pada kolom $|Fa(x_i) - Fe(x_i)|$ dibandingkan dengan nilai dari tabel Kolmogorov-Smirnov dengan jumlah data 20 dan $\alpha = 0,05$. Pada kasus ini nilai hitung adalah 0,27 dan nilai dari tabel adalah 0,294 sehingga kesimpulan bahwa data tersebut terdistribusi secara *uniform* diterima.

Untuk cara pembangkitan data yang terdistribusi secara *uniform* dapat digunakan Persamaan 2.8. Untuk mendapatkan nilai x_i dibutuhkan dua buah parameter yaitu nilai minimum (a) dan nilai maksimum (b). Pada Tabel 2.3, a bernilai 11 dan b bernilai 21. r adalah nilai acak yang terdistribusi secara *uniform* dengan nilai minimal 0 dan maksimal 1.

$$x_i = a + (b - a) \times r \quad (2.8)$$

Tabel 2.3 Contoh Jumlah Kendaraan Dengan Distribusi Uniform

x_i	Frekuensi x_i	Frekuensi Relatif x_i	$Fa(x_i)$	$Fe(x_i)$	$ Fa(x_i) - Fe(x_i) $
11	5	0,25	0,25	0,52	0,27
12	2	0,1	0,35	0,57	0,22
13	1	0,05	0,4	0,62	0,22
14	2	0,1	0,5	0,67	0,17
15	2	0,1	0,6	0,71	0,11
16	2	0,1	0,7	0,76	0,06
17	1	0,05	0,75	0,81	0,06
20	2	0,1	0,85	0,95	0,1
21	3	0,15	1	1	0

2.1.1 Pembangkitan Data dengan Distribusi Normal

Data yang akan dibangkitkan dengan mengikuti pola distribusi normal menggunakan Persamaan 2.2 dan Persamaan 2.3. Pada Tabel 2.3 dapat dilihat contoh data yang dibangkitkan. Kolom r_1 dan r_2 adalah angka yang dibangkitkan secara acak dengan mengikuti pola distribusi *uniform* dengan batas bawah 0 dan batas atas 1. Kolom U didapat dengan menggunakan Persamaan 2.2 dan kolom V , hasil pembangkitan bilangan dengan distribusi normal, didapat dengan menggunakan Persamaan 2.3. Hasil yang didapat merupakan sekumpulan bilangan dengan rata-rata dan standar deviasi yang telah diinginkan, dalam kasus ini adalah 5 dan 2,5. Rata-rata dan standar deviasi dapat berupa sebuah bilangan real.

Tabel 2.4 Contoh pembangkitan dengan distribusi normal

r_1	r_2	U	x_i
0,300550	0,112111	0,661732	6,654330
0,181495	0,441102	0,440338	6,100846
0,760908	0,361614	0,372202	5,930506
0,148968	0,774337	-1,271000	1,822500
0,063477	0,618574	-1,049190	2,377025
0,478648	0,376462	0,560452	6,401130
0,622406	0,722937	-0,632500	3,418745
0,852948	0,297653	0,355156	5,887889

2.1.2 Pembangkitan Data dengan Distribusi Eksponensial

Pada subbab ini akan dibahas cara pembangkitan data dengan mengikuti distribusi eksponensial beserta contohnya. Untuk membangkitkan data dengan distribusi eksponensial hanya dibutuhkan sebuah parameter, yaitu rata-rata. Rata-rata yang akan digunakan sebagai dasar pembangkitan pada Tabel 2.4 adalah 5. Kolom pertama adalah data yang dibangkitkan secara acak dengan distribusi *uniform* dan dengan rentang 0 sampai 1. Kolom kedua, berisi hasil pembangkitan dengan distribusi eksponensial. Hasil ini didapat dengan memasukkan hasil dari kolom pertama kedalam Persamaan 2.7.

Tabel 2.5 Contoh pembangkitan dengan distribusi eksponensial

rl	x_i
0,890711	0,251317
0,026729	7,865052
0,551725	1,291388
0,182052	3,699018
0,617176	1,047955
0,117911	4,642227
0,832193	0,398881

2.1.3 Pembangkitan Data dengan Distribusi Uniform

Cara pembangkitan data dengan distribusi *uniform* dengan rentang yang telah ditentukan pada dasarnya adalah memperlebar skala dari bilangan acak dengan rentang awal 0 sampai 1. Cara untuk memperlebar skala tersebut dapat menggunakan Persamaan 2.8. Hasil dari contoh pembangkitan bilangan acak dengan distribusi *uniform* dan dengan rentang 5 sampai 10 dapat dilihat pada Tabel 2.5. Kolom pertama pada tabel tersebut adalah nilai acak *uniform* dengan rentang 0 sampai 1 dan kolom kedua adalah nilai acak *uniform* dengan rentang 5 sampai 10.

rl	V
0,887352	9,436760
0,636404	8,182021
0,387234	6,936168
0,847379	9,236895
0,862860	9,314299
0,668519	8,342596
0,701255	8,506277

Untuk cara pembangkitan bilangan acak *uniform* dengan rentang 0 sampai 1, dalam Tugas Akhir ini digunakan bantuan perangkat lunak Visual Studio 2010.

2.2 Jaringan Saraf Tiruan

Cabang ilmu kecerdasan buatan cukup luas, dan erat kaitannya dengan disiplin ilmu yang lainnya. Hal ini bisa dilihat dari berbagai aplikasi yang merupakan hasil kombinasi dari berbagai ilmu. Seperti halnya yang ada pada peralatan medis yang berbentuk aplikasi. Sudah berkembang bahwa aplikasi yang dibuat merupakan hasil perpaduan dari ilmu kecerdasan buatan dan juga ilmu kedokteran atau lebih khusus lagi yaitu ilmu biologi

Jaringan Saraf Tiruan [5] merupakan kategori ilmu *Soft Computing*. Jaringan Saraf Tiruan sebenarnya mengadopsi dari kemampuan otak manusia yang mampu memberikan stimulasi/rangsangan, melakukan proses, dan memberikan keluaran. Keluaran diperoleh dari variasi stimulasi dan proses yang terjadi di dalam otak manusia. Kemampuan manusia dalam memproses informasi merupakan hasil kompleksitas proses di dalam otak. Misalnya, yang terjadi pada anak-anak, mereka mampu belajar untuk melakukan pengenalan meskipun mereka tidak mengetahui algoritma apa yang digunakan. Kekuatan komputasi yang luar biasa dari otak manusia ini merupakan sebuah keunggulan di dalam kajian ilmu pengetahuan.

Dalam Tugas Akhir ini akan digunakan dua jenis Jaringan Saraf Tiruan yaitu Jaringan Saraf Tiruan Multi Layer Perceptron dan Jaringan Saraf Tiruan Regresi Linear. Kesamaan dari kedua model ini adalah menggunakan lapisan masukan dan jenis laju pembelajaran yang sama.

2.2.1 Lapisan Masukan

Seperti yang telah diketahui, data jumlah kendaraan yang akan digunakan adalah jumlah kendaraan permenit yang telah dijumlahkan dalam rentang waktu 15 menit. Jumlah *neuron* yang digunakan sebagai data masukan untuk lapisan [5] ini akan bervariasi dengan jumlah maksimal adalah 6. Semua data masukan akan dipaksa kedalam rentang 0 sampai 1 dengan membagi

masing-masing data dengan nilai yang paling besar. Misalnya sebuah dataset data masukan berisi angka 1,2,3,4 maka setelah dinormalkan menjadi 0,25;0,5;0,75;1.

Kombinasi dari berbagai jenis data masukan akan digunakan dalam uji coba untuk mendapatkan hasil yang paling optimal. Data masukan kesatu sampai dengan kelima akan berisi jumlahan data kendaraan selama t , dimana $t=15$ menit. Data masukan keenam adalah jenis hari, kategori 1 dan kategori 2.

Data yang termasuk dalam kategori 1 adalah hari aktif kerja Senin, Selasa, Rabu, Kamis, dan Jumat. Sabtu dan Minggu serta hari libur lainnya akan dikategorikan dalam kategori 2. Jika dalam suatu percobaan jenis hari dianggap berpengaruh maka data masukan akan bernilai satu jika hari diwaktu yang akan diramalkan adalah berada dalam kategori 1 dan 0 jika sebaliknya. Namun, jika jenis hari dianggap tidak berpengaruh maka baik hari-hari yang berada dikategori 1 maupun hari-hari yang berada dikategori 2 akan diberi nilai 0.

Salah satu contoh kombinasi data masukan adalah sebagai berikut:

Data masukan yang pertama adalah data jumlah kendaraan satu waktu sebelum waktu yang diinginkan atau dengan kata lain sama dengan jumlahan data jumlah kendaraan selama 15 menit sebelumnya.

Masukan yang kedua adalah data jumlah kendaraan dua waktu sebelumnya atau dengan kata lain jumlahan jumlah kendaraan mulai dari 15 menit sampai 30 menit sebelumnya.

Masukan yang ketiga dan kelima masing-masing diambil dari jumlahan jumlah kendaraan selama 15 menit dijam yang sama dengan waktu observasi hanya saja dihari sebelumnya dan 7 hari sebelumnya.

Masukan yang keempat adalah jumlahan data kendaraan selama 15 menit yang terjadi dihari sebelumnya dan dengan selisih 15 menit.

Masukan yang terakhir adalah jenis hari, jika pada hari yang diobservasi jenis harinya adalah hari kerja(Senin, Selasa, Rabu,

Kamis, Jumat) maka data ini akan berisi 0 dan jika hari yang diobservasi adalah akhir pekan atau hari libur maka data ini akan berisi 1.

Dapat dikatakan dengan singkat, jika waktu observasi adalah t , maka input pertama sampai input kelima adalah :

1. $t-1$
2. $t-2$
3. $t-96$
4. $t-97$
5. $t-(96 \times 7)$

Karena jumlahan kendaraan telah dikelompokkan setiap 15 menit maka akan ada 96 kelompok data setiap harinya. Untuk representasi dalam bentuk gambar dengan kombinasi data masukan seperti diatas dapat dilihat pada Gambar 2.1 untuk Jaringan Saraf Tiruan Multi Layer Perceptron dan Gambar 2.2 untuk Jaringan Saraf Tiruan Regresi Linear

2.2.2 Laju Pembelajaran

Batas yang akan digunakan sebagai laju pembelajaran [5] adalah 0.001 sampai 0.5. Terlalu besarnya laju pembelajaran dapat membuat tidak tercapainya hasil optimum namun kecilnya laju pembelajaran dapat memperbanyak jumlah perulangan sebelum mencapai hasil optimum. Dalam tahap uji coba akan dicobakan banyak nilai laju pembelajaran untuk mendapatkan nilai yang paling sesuai.

2.3 Korelasi

Korelasi [6] adalah salah satu teknik statistik yang digunakan untuk mencari hubungan dua variabel atau lebih yang sifatnya kuantitatif. Nilai korelasi berada diantara -1 sampai 1 semakin dekat dengan 0 berarti hampir tidak ada keterkaitan antar kedua variabel tersebut dan semakin dekat dengan 1 berarti kedua variabel tersebut saling terkait dengan kuat. Nilai negatif mengindikasikan bahwa hubungan kedua variabel tersebut adalah saling keterbalikan.

Dalam Tugas Akhir ini penggunaan korelasi dimanfaatkan sebagai parameter bantuan saat akan melakukan inisiasi bobot pada metode Jaringan Saraf Tiruan Regresi Linear. Korelasi antara dua buah variabel dihitung dengan Menggunakan Persamaan 2.9.

$$r = \frac{\sum_{i=0}^k (x_i - m_1)(y_i - m_2)}{\sqrt{\sum_{i=0}^k (x_i - m_1)^2 \sum_{i=0}^k (y_i - m_2)^2}} \quad (2.9)$$

r adalah hasil dari korelasi antara dua buah kelas (x dan y) dan k adalah jumlah anggota dalam kedua kelas. m_1 adalah rata-rata dari kelas x dan m_2 adalah rata-rata dari kelas y .

Misalnya terdapat dua buah kelas x dan y . x beranggotakan 5,3,5,6,8 dan y beranggotakan 4,7,2,8,5. Oleh karena anggota dari masing-masing kelompok adalah 5 k adalah 5. Rata-rata dari kelas x adalah 5,4 dan rata-rata kelas y adalah 5,2. Korelasi dari kedua buah kelas diatas setelah dihitung dengan Persamaan 2.5 adalah $-0,0807$.

2.4 Jaringan Saraf Tiruan Multi Layer Perceptron

Jaringan Saraf Tiruan Multi Layer Perceptron [5] adalah salah satu tipe dari Jaringan Saraf Tiruan yang mempunyai sejumlah *neuron* pada lapisan tersembunyi yang dihubungkan oleh bobot-bobot penghubung. *Neuron-neuron* tersebut disusun dalam lapisan-lapisan yang terdiri dari satu lapisan masukan (*input layer*), satu lapisan tersembunyi (*hidden layer*), dan satu lapisan keluaran (*output layer*).

Input Layer atau bisa juga disebut sebagai lapisan masukan adalah lapisan yang dimana mempunyai jumlah neuron yang sama dengan jumlah *input*. Data yang menjadi masukan pada lapisan ini juga menjadi keluaran dari lapisan ini juga atau dengan kata lain tidak ada fungsi aktivasi pada lapisan ini.

Jumlah lapisan tersembunyi yang dapat diimplementasi dalam *Jaringan Saraf Tiruan Multi Layer Perceptron* adalah satu

sampai dengan tak hingga. Setiap lapisan tersembunyi dibuat dengan tujuan merepresentasikan suatu hal tertentu. Dalam Tugas Akhir ini hanya akan ada satu buah lapisan tersembunyi dengan jumlah *neuron* yang sama dengan x . Nilai x didapat dengan menggunakan Persamaan 2.10 [7].

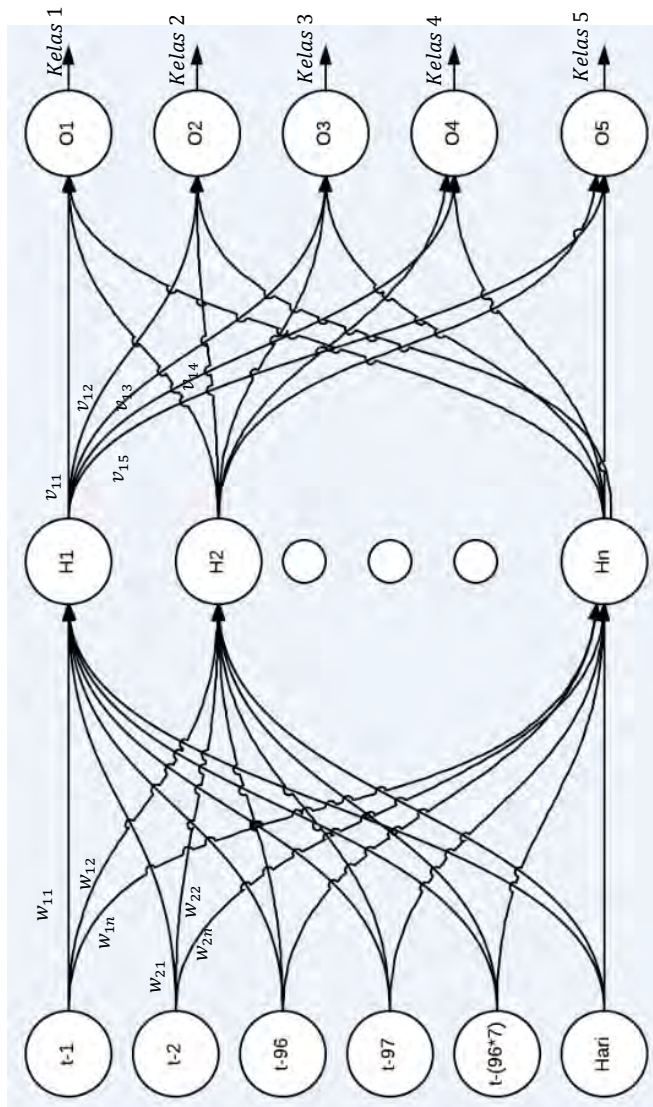
$$x = \sqrt{(q + 2)p} + 2 \sqrt{\frac{p}{q + 2}} \quad (2.10)$$

q adalah jumlah jumlah *neuron* pada lapisan keluaran dan p adalah jumlah *neuron* pada lapisan masukan. Misalnya pada lapisan masukan ada 6 buah *neuron* dan pada lapisan keluaran ada 5 *neuron* maka jumlah *neuron* yang tepat pada lapisan tersembunyi adalah 8.

Output Layer atau lapisan keluaran adalah lapisan yang berada di paling akhir. Jumlah *neuron* pada lapisan ini akan berjumlah satu jika hanya ada dua kelas pada data asli, atau sama dengan jumlah kelas pada asli jika terdapat lebih dari dua kelas pada data asli.

Struktur Jaringan Saraf Tiruan Multi Layer Perceptron dapat dilihat pada Gambar 2.3. Data masukan yang digunakan pada Gambar 2.3 sama dengan yang telah dicontoh pada subbab 2.2.1. Keluaran yang dihasilkan adalah lima buah kelas(kelas 1 sampai kelas 5) yang menandakan tingkat kepadatan kendaraan. Kelas 1 berarti kelas jumlah kendaraan yang sangat tidak padat sampai Kelas 5 berarti kelas jumlah kendaraan yang sangat padat.

Pada Gambar 2.3 w_{ij} adalah simbol untuk bobot yang menghubungkan lapisan masukan sampai lapisan tersembunyi dan v_{ij} adalah bobot yang menghubungkan lapisan tersembunyi dengan lapisan keluaran.



Gambar 2.3 Struktur Jaringan Saraf Tiruan Multi Layer Perceptron

Proses penting yang harus dilakukan terlebih dahulu sebelum membuat jaringan adalah menginisiasi bobot. Dalam proses pengerjaan metode Jaringan Saraf Tiruan Multi Layer Perceptron, akan dilakukan banyak uji coba untuk inisiasi bobot. Bobot yang diinisiasi adalah bobot dari lapisan masukan sampai lapisan tersembunyi, lapisan tersembunyi sampai lapisan keluaran dan nilai bias disetiap neuron pada lapisan tersembunyi. Rentang nilai yang akan diuji cobakan akan bervariasi, mulai dari -1 sampai 1. Dalam setiap percobaan rentang ini akan diubah-ubah namun dengan batas paling rendah -1 dan paling tinggi 1. Akan ada juga percobaan dimana semua bobot awal diinisiasi sama dengan 0. Antara bobot yang berada diantara lapisan masukan sampai lapisan tersembunyi akan diinisiasi secara terpisah dengan dua jenis bobot lainnya sehingga rentang inisiasi antara satu jenis bobot dengan yang lainnya dapat berbeda-beda.

Jaringan dilatih agar keluaran jaringan sesuai dengan pola pasangan masukan dan target yang telah ditentukan. Proses pelatihan adalah proses iteratif untuk menentukan bobot-bobot koneksi antara neuron yang paling optimal. Kata *backpropagation* yang sering dikaitkan pada MLP merujuk pada cara bagaimana gradien perubahan bobot dihitung. Jaringan MLP yang sudah dilatih dengan baik akan memberikan keluaran yang masuk akal jika diberi masukan yang serupa (tidak harus sama) dengan pola yang dipakai dalam pelatihan.

Berikut ini adalah tahap-tahapan dalam penyelesaian masalah menggunakan metode Jaringan Saraf Tiruan Multi Layer Perceptron:

1. Memilih Data masukan

Tahapan memilih data masukan adalah tahapan dimana pengguna memilih apa saja yang akan dijadikan data masukan, seperti yang telah dijelaskan sebelumnya tentang data masukan yang dapat berupa beberapa kombinasi. Karena kombinasi data masukan sangat menentukan tingkat keakuratan akurasi maka pada tahapan uji coba akan

dilakukan banyak percobaan kombinasi untuk mencari kombinasi yang paling optimal.

2. Menentukan Jumlah Iterasi Maksimum

Pada tahapan ini pengguna menentukan jumlah iterasi maksimal dan jumlah *neuron* pada lapisan tersembunyi. Jumlah iterasi maksimal yang besar akan berpotensi tinggi meningkatkan akurasi sistem, namun ada batas dimana setelah jumlah iterasi melewati batas tersebut tidak ada perubahan yang signifikan pada akurasi. Seperti halnya kombinasi pada data masukan, jumlah neuron pada lapisan tersembunyi juga mempunyai jumlah optimal yang hanya bisa didapat dengan melakukan percobaan. Pada tahap ini dapat ditentukan berapa jumlah neuron pada lapisan tersembunyi yang diinginkan.

3. Inisiasi Bobot

Tahapan selanjutnya yang harus dilakukan adalah menginisai bobot dengan langkah- langkah seperti yang telah dipaparkan pada bagian sebelumnya. Semua jenis bobot, mulai dari bobot antara neuron pada lapisan masukan dan lapisan tersembunyi, antara neuron pada lapisan tersembunyi dan lapisan keluaran dan bobot pada setiap neuron di lapisan tersembunyi (bias), diinisiasi pada langkah ini. Penghitung iterasi dimulai dari 0 sampai dengan jumlah iterasi maksimal dikurangi 1.

4. Klasifikasi

Tahapan klasifikasi adalah tahapan dimana menghitung semua nilai dari semua *neuron*. Untuk menghitung nilai pada *neuron* yang berada di lapisan tersembunyi digunakan Persamaan 2.11.

$$H_k = \sum_{i=1}^q V_{ik} I_i \quad (2.11)$$

H_j adalah *neuron* indeks ke j pada lapisan tersembunyi. V_{ik} adalah bobot yang menghubungkan *neuron* ke k pada lapisan tersembunyi dan *neuron* ke i pada lapisan masukan. I_i adalah

neuron ke i pada lapisan masukan. q adalah jumlah *neuron* pada lapisan masukan.

Neuron yang berada pada lapisan keluaran dihitung dengan Persamaan 2.12. T_j adalah *neuron* ke j pada lapisan keluaran, W_{jk} adalah bobot yang menghubungkan *neuron* ke j pada lapisan keluaran dan *neuron* ke k pada lapisan tersembunyi. p adalah jumlah *neuron* pada lapisan tersembunyi.

$$T_j = \sum_{k=1}^p W_{jk} H_k \quad (2.12)$$

Setelah melakukan perhitungan, dari kelima *neuron* yang berada pada lapisan keluaran dipilih nilai yang terbesar. Misalnya nilai yang dihasilkan oleh kelima *neuron* pada lapisan keluaran 0,1;0,2;0,3;0,4 dan 0,5, maka data tersebut digolongkan kedalam kelas ke lima.

5. Mencari Nilai Kesalahan

Jika hasil klasifikasi benar, maka tidak akan ada proses memperbaiki bobot, namun jika hasil klasifikasi salah, maka akan dilakukan proses pembaruan bobot. Proses pembaruan bobot dilakukan dengan terlebih dahulu menghitung nilai kesalahan pada lapisan keluaran. Nilai kesalahan pada lapisan keluaran dilambangkan dengan simbol Err_j dimana j melambangkan indeks dari data yang ingin diramalkan. Err_j didapat dengan mengalikan hasil aktual dengan 1 dikurangi hasil aktual dan hasil ramalan dikurangi dengan hasil aktual. Err_j juga dapat ditulis dalam bentuk lain seperti pada Persamaan 2.13. T_j adalah kelas hasil ramalan.

$$Err_j = (1 - O_j)(T_j - O_j). \quad (2.13)$$

Setelah mendapatkan nilai kesalahan pada lapisan keluaran maka akan dilanjutkan dengan mencari nilai kesalahan pada lapisan tersembunyi. Kesalahan pada lapisan tersembunyi, dilambangkan dengan simbol Err_h , Err_h dapat dihitung dengan menggunakan Persamaan 2.14.

$$Err_h = O_j(1 - O_j) \sum_k Err_k W_{jk} \quad (2.14)$$

O_j adalah hasil keluaran yang sebenarnya dan W_{jk} adalah nilai bobot yang menghubungkan antara *neuron* pada lapisan tersembunyi dan keluaran.

6. Memperbaiki Bobot

Langkah selanjutnya yang dilakukan setelah mendapatkan nilai kesalahan pada lapisan tersembunyi adalah memperbaharui bobot. Bobot dan bias diperbaharui dengan menambahkan nilai bobot atau bias yang lama dengan laju pembelajaran dikalikan dengan nilai kesalahan pada lapisan tersembunyi dan nilai keluaran aktual. Untuk pembaruan pada bobot yang menghubungkan lapisan masukan dan lapisan tersembunyi juga dilakukan dengan cara yang sama.

Kesimpulannya adalah metode ini cocok digunakan untuk menyelesaikan masalah yang tidak linier, yang tidak dapat dimodelkan secara matematis. Jaringan cukup belajar dari pasangan data masukan dan target yang diinginkan, setelah itu jaringan dapat mengenali pola yang mirip dengan masukan ketika dilakukan pelatihan. Karena itu, kinerja jaringan pun ditentukan oleh banyaknya pasangan dataset selama pembelajaran.

Bila data pembelajaran cukup banyak dan konsisten, akurasi jaringan akan tinggi, sebaliknya bila data pembelajaran tidak memadai, akurasi jaringan rendah. Selain data pembelajaran, akurasi jaringan juga ditentukan oleh pemilihan topologi yang tepat.

Proses pembentukan jaringan sangat melelahkan, dilakukan secara terus menerus hingga diperoleh jaringan yang paling baik. Tetapi setelah jaringan yang optimal ditemukan, proses pengenalan pola dapat dilakukan secara cepat, lebih cepat bila dibandingkan metoda lainnya.

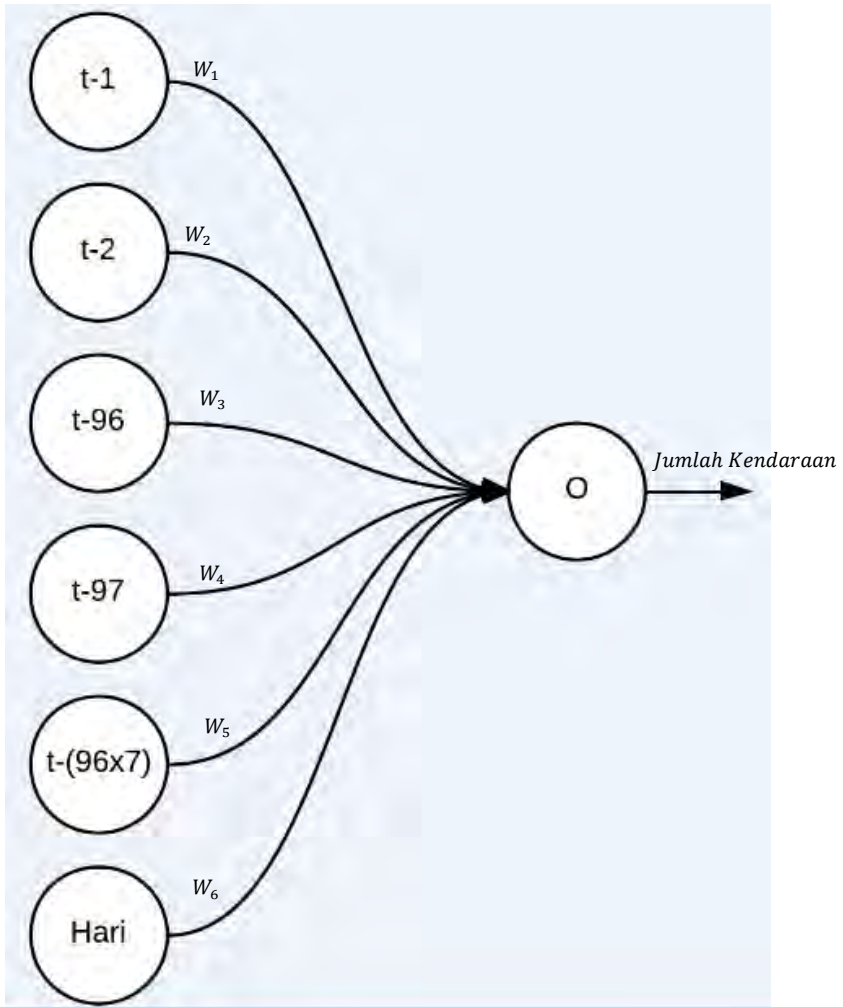
2.5 Jaringan Saraf Tiruan Regresi Linear

Secara umum, analisis regresi pada dasarnya adalah studi mengenai ketergantungan satu variabel dependen (terikat) dengan satu atau lebih variabel independen (bebas), dengan tujuan untuk mengestimasi atau nilai rata-rata variabel dependen berdasarkan nilai variabel independen yang diketahui. Pusat perhatian adalah pada upaya menjelaskan dan mengevaluasi hubungan antara suatu variabel dengan satu atau lebih variabel independen.

Hasil analisis regresi adalah berupa koefisien regresi untuk masing-masing variabel independen. Koefisien ini diperoleh dengan cara memprediksi nilai variabel dependen dengan suatu persamaan. Koefisien regresi dihitung dengan dua tujuan sekaligus pertama, meminimumkan penyimpangan antara nilai aktual dan nilai estimasi variabel dependen, kedua, mengoptimalkan korelasi antara nilai aktual dan nilai estimasi variabel dependen berdasarkan data yang ada.

Dalam Jaringan Saraf Tiruan Regresi Linear sendiri, metode Regresi Linear digunakan untuk mengoptimalkan bobot dari lapisan tersembunyi menuju ke lapisan keluaran [8] karena hasil dari lapisan keluaran adalah terikat kepada keluaran dari lapisan tersembunyi. Struktur dari Jaringan Saraf Tiruan Regresi Linear dapat dilihat pada Gambar 2.4. Seperti pada Gambar 2.1, pada Gambar 2.4 contoh data masukan yang digunakan adalah sama seperti yang telah diberikan pada contoh pada subbab 2.2.1.

Tidak ada lapisan tersembunyi pada Jaringan Saraf Tiruan Regresi Linear oleh karena itu bobot yang digunakan hanya satu jenis yaitu bobot yang menghubungkan antara lapisan masukan dengan lapisan keluaran, pada Gambar 2.4 bobot ini ditunjukkan dengan symbol W_1 sampai W_6 .



Gambar 2.4 Struktur Jaringan Saraf Tiruan Regresi Linear

Berikut adalah contoh tahapan pengerjaan Jaringan Saraf Tiruan Regresi Linear:

1. Memilih Data Masukan dan Mementukan Iterasi Maksimum.

Tahapan ini dikerjakan dengan langkah-langkah yang sama dengan metode Jaringan Saraf Tiruan Multi Layer Perceptron. Misalkan data pembelajaran adalah sebuah dataset T dimana $T = [\{ (x_1^1, x_1^2, \dots, x_1^6), a_1 \}, \{ (x_2^1, x_2^2, \dots, x_2^6), a_2 \}, \dots, \{ (x_m^1, x_m^2, \dots, x_m^6), a_m \}]$. x_i^1 sampai x_i^6 adalah data vektor data masukan pada indeks ke- i , terdapat m buah data masukan pada dataset T , a_i adalah data keluaran pada indeks ke- i pada dataset T . Representasi dari dataset masukan dapat dilihat pada Persamaan 2.15.

$$x = \begin{pmatrix} x_1^1 & x_n^1 \\ x_1^2 & x_n^2 \\ x_1^3 & x_n^3 \\ \dots & \dots \\ x_1^m & x_n^m \end{pmatrix} \quad (2.15)$$

2. Inisiasi Bobot

Proses inisiasi bobot dalam metode Jaringan Saraf Tiruan Regresi Linear akan dilakukan dengan mempertimbangkan nilai korelasi. Korelasi yang akan dihitung adalah korelasi antara setiap input dengan output pada data latihan. Dalam Tugas Akhir ini penghitungan korelasi dilakukan dengan tujuan membantu inisiasi bobot. Semakin tinggi nilai korelasi antara sebuah data masukan dengan data keluaran maka semakin tinggi juga pemberian bobot data masukan tersebut.

Nilai korelasi setiap pasang data masukan dan data keluaran dapat berada diantara 0 sampai 1, karena jumlah data masukan pada lapisan masukan dapat berjumlah paling banyak lima maka proses inisiasi bobot untuk setiap data masukan akan diskalakan dengan total jumlah sama dengan 1. Setiap bobot akan diinisiasi dengan nilai yang sama dengan

nilai korelasi data masukan tersebut dibagi dengan total seluruh nilai korelasi.

3. Menghitung Nilai Prediksi

Nilai prediksi didapatkan dengan mengalihkan semua data masukan dengan bobotnya masing-masing. Nilai yang didapatkan akan dikalikan dengan data terbesar dalam kelompok data masukan untuk kemudian menjadi hasil ramalan. Persamaannya dapat dilihat pada Persamaan 2.16. H_j adalah hasil ramalan pada data ke- j dan n adalah jumlah *neuron* pada lapisan masukan.

$$H_j = \sum_{i=1}^n W_i x_j^i \quad (2.16)$$

W adalah matriks bobot yang mempunyai jumlah baris sejumlah *neuron* yang ada lapisan masukan dan hanya 1 kolom karena hanya ada 1 *neuron* di lapisan keluaran. Pada Tugas Akhir ini matriks W akan berdimensi 1x6.

4. Memperbaiki Bobot

Untuk setiap iterasi dilakukan perbaikan bobot dengan menggunakan Persamaan 2.17 [8] untuk setiap nilai j . j akan berisi nilai dari 1 sampai 6 yang melambangkan banyaknya jumlah *neuron* di lapisan masukan. W_j^1 adalah bobot pada iterasi sebelumnya. γ adalah laju pembelajaran, detail mengenai laju pembelajaran dapat dilihat pada subbab 2.2.2.

$$W_j = W_j^1 - \gamma \left(a_1 - \sum_{i=1}^n w_i x_i^1 \right) x_j^1 - \dots - \left(a_m - \sum_{i=1}^n w_i x_i^m \right) x_j^m \quad (2.17)$$

BAB III

PERANCANGAN PERANGKAT LUNAK

Perancangan merupakan bagian penting dari pembuatan suatu perangkat lunak yang berupa perencanaan-perencanaan secara teknis aplikasi yang dibuat. Sehingga bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Berawal dari deskripsi umum aplikasi hingga perancangan proses.

3.1 Deskripsi Umum Sistem

Dalam Tugas Akhir kali ini akan mengimplementasikan metode Jaringan Saraf Tiruan Multi Layer Perceptron dan Jaringan Saraf Tiruan Regresi Linear. Kedua metode tersebut digunakan untuk melakukan peramalan hanya saja dengan metode Jaringan Saraf Tiruan Multi Layer Perceptron peramalan yang dilakukan dalam bentuk klasifikasi dan pada Jaringan Saraf Tiruan Regresi Linear hasil yang dihasilkan adalah angka jumlah kendaraan secara numerik. Setelah mendapatkan hasilnya parameter yang digunakan untuk menghitung tingkat kesuksesan dari Jaringan Saraf Tiruan Multi Layer Perceptron adalah prosentasi tingkat kesamaan jumlah kelas yang diprediksi sedangkan pada Jaringan Saraf Tiruan Regresi Linear tingkat keberhasilannya dinilai dengan nilai dari *Mean Absolute Percentage Error*.

3.2 Perancangan Pembangkitan Data

Pembangkitan data pada perangkat lunak digunakan dengan mengikuti persamaan-persamaan pada subbab 2.2. Untuk setiap rentang waktu yang telah ditetapkan jumlah kendaraan akan terdistribusi secara normal, *uniform* dan eksponensial. *Pseudocode* untuk distribusi normal dapat dilihat pada Tabel 3.2, *pseudocode* distribusi *uniform* pada Tabel 3.3 dan *pseudocode* distribusi eksponensial pada Tabel 3.4. Daftar symbol-simbol yang digunakan pada ketiga tabel tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar Simbol dan Keterangan

Daftar Simbol	Keterangan
r	Variabel acak <i>uniform</i> antara 0 sampai 1
$randStdNormal$	Variabel acak normal antara 0 sampai 1
$randNormal$	Variabel acak normal dengan rata-rata dan standar deviasi yang telah ditentukan
$randUniform$	Variabel acak <i>uniform</i> dengan rentang yang telah ditentukan
$randEksponensial$	Variabel acak eksponensial
μ	Rata-rata
σ	Standar Deviasi
a	Nilai minimum pada distribusi uniform
b	Nilai maksimum pada distribusi uniform

Tabel 3.2 *Pseudocode* Pembangkitan Data dengan Distribusi Normal

Pseudocode pembangkitan data dengan distribusi normal
$r_1 \leftarrow rand(0,1)$ $r_2 \leftarrow rand(0,1)$ $randStdNormal \leftarrow \sqrt{-2 \times \log r_1} \times \sin(2 \times \pi \times r_2)$ $randNormal = \mu + (\sigma \times randStdNormal)$

Tabel 3.3 *Pseudocode* Pembangkitan Data dengan Distribusi Uniform

Pseudocode pembangkitan data dengan distribusi uniform
$r \leftarrow rand(0,1)$ $randUniform = a + (b - a) \times r$

Tabel 3.4 *Pseudocode* Pembangkitan Data dengan Distribusi Eksponensial

Pseudocode pembangkitan data dengan distribusi eksponensial
$r \leftarrow \text{rand}(0,1)$ $\text{randEksponensial} \leftarrow -1 \times \log(1 - r) \times \mu$

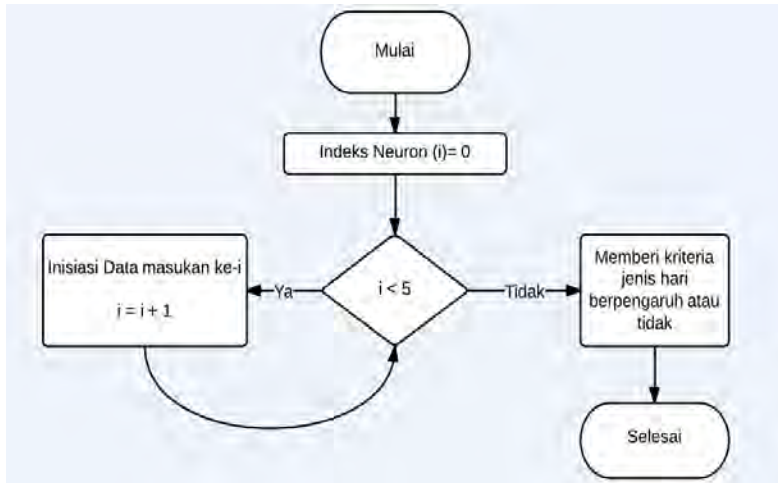
3.3 Diagram Alir Jaringan Saraf Tiruan Multi Layer Perceptron

3.3.1 Data masukan

Tahapan pertama yang harus dilewati oleh pengguna adalah menentukan kriteria data masukan. Untuk data masukan pertama sampai kelima, kriteria yang dimasukan oleh pengguna adalah untuk data masukan tersebut data jumlahan kendaraan berapa waktu sebelumnya yang diinginkan. Seperti yang telah dijelaskan, 1 satuan waktu dalam sistem ini adalah jumlahan kendaraan selama 15 menit. Data masukan keenam akan mengindikasikan apakah jenis hari(libur atau tidak) ikut mempengaruhi atau tidak. Diagram alir untuk proses membuat kriteria data masukan dapat dilihat pada Gambar 3.1 dimana i adalah variabel bantuan untuk melakukan iterasi.

3.3.2 Inisiasi Bobot

Pada Gambar 3.2 dapat dilihat proses inisiasi bobot yang digambarkan dengan diagram alir. Bobot adalah variabel yang berdimensi 2 dengan dimensi pertama sama dengan n dan dimensi kedua sama dengan m . Variabel i dan j adalah variabel yang digunakan sebagai pendanda iterasi, dimana i untuk membantu melakukan perulangan sebanyak n kali dan j digunakan untuk melakukan perulangan sebanyak m kali. Ada 3 jenis bobot yaitu pada hubungan antara setiap *neuron* pada lapisan masukan dengan *neuron* pada lapisan tersembunyi, bobot yang berada diantara lapisan tersembunyi dan lapisan keluaran dan bobot pada setiap *neuron* pada lapisan tersembunyi yang juga dapat disebut bias.



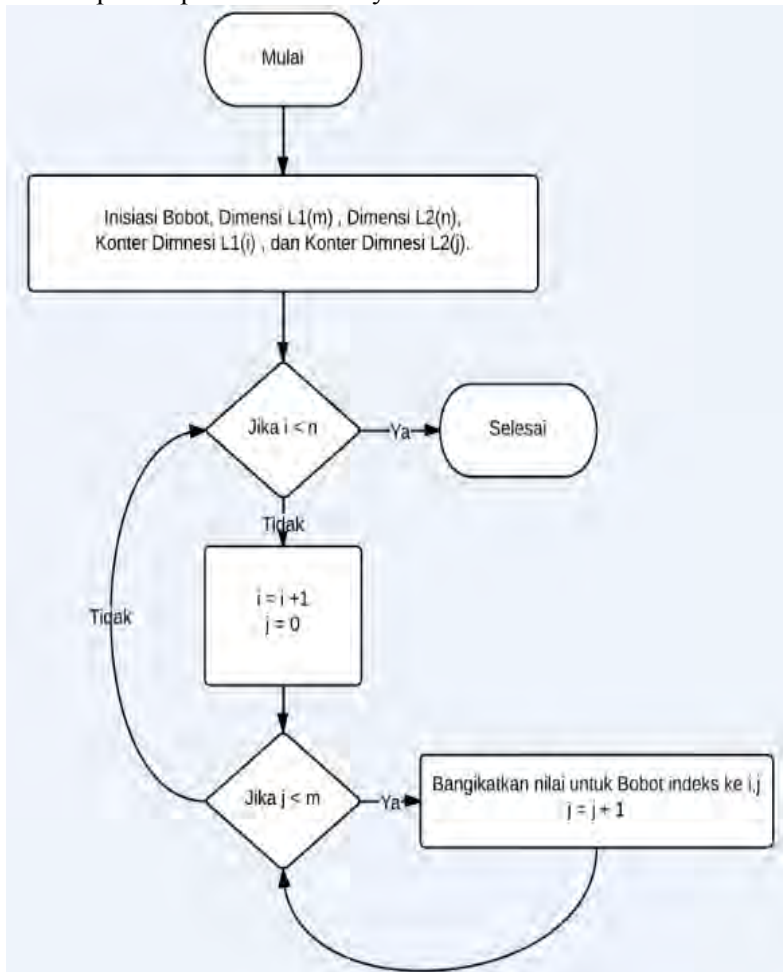
Gambar 3.1 Diagram Alir Memberi Kriteria Data Masukan

Proses inisiasi yang dimaksud pada diagram alir pada Gambar 3.2 adalah untuk variabel Bobot maka harus disiapkan tempat dua dimensi dengan besar n kali m sedangkan untuk n dan m akan diberikan angka yang sesuai dengan besar dimensi Bobot yang diinginkan, sebaliknya untuk variabel i dan j yang berfungsi sebagai penghitung akan diberi nilai 0.

Jika variabel Bobot digunakan untuk menginisiasi bobot dari lapisan masukan menuju lapisan tersembunyi maka nilai n akan sama dengan 6(jumlah *neuron* pada lapisan masukan) dan m akan bernilai jumlah *neuron* pada lapisan tersembunyi. Jika variabel bobot yang diinginkan adalah bobot dari lapisan tersembunyi sampai lapisan keluaran maka n akan sama dengan jumlah *neuron* pada lapisan tersembunyi dan m akan sama dengan 5(jumlah *neuron* pada lapisan keluaran).

Jenis bobot yang berbeda dari kedua bobot lainnya adalah bobot untuk bias yang berada pada lapisan tersembunyi. Jika yang diinginkan adalah bobot ini maka n akan bernilai sama dengan

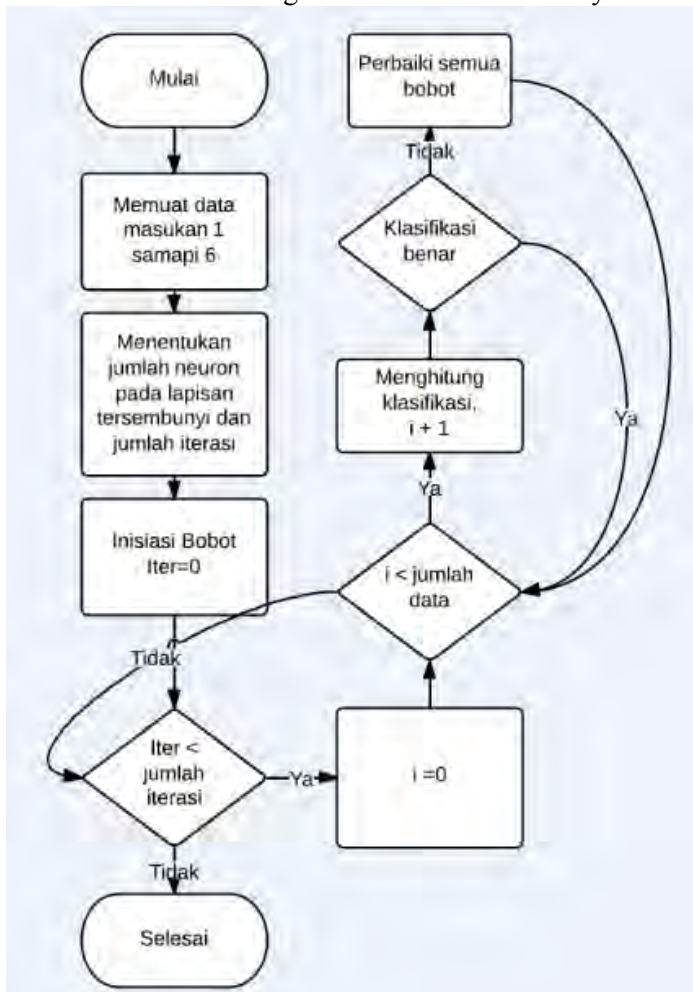
1(jumlah lapisan tersembunyi) dan m akan bernilai sama dengan *neuron* pada lapisan tersembunyi.



Gambar 3.2 Diagram Alir inisiasi bobot Jaringan Saraf Tiruan Multi Layer Perceptron

3.3.3 Pelatihan

Pada Gambar 3.3 dapat dilihat secara umum tahapan pengerjaan dari metode Jaringan Saraf Tiruan Multi Layer



Gambar 3.3 Diagram Alir Jaringan Saraf Tiruan Multi Layer Perceptron

Perceptron. Alur pengerjaannya dimulai dari “mulai” sampai dengan “selesai”. Proses berhenti ketika jumlah iterasi telah sampai pada jumlah iterasi maksimal yang telah ditetapkan.

Tahapan memuat data masukan dilakukan dengan mengikuti langkah-langkah yang ditunjukkan pada digram alir yang ditunjukkan oleh Gambar 3.1.

Pada tahapan menentukan jumlah *neuron* pada lapisan tersembunyi digunakan bantuan Persamaan 2.1. Dengan jumlah *neuron* pada lapisan masukan 6 dan jumlah *neuron* pada lapisan keluaran 5 maka didapatkan jumlah *neuron* pada lapisan tersembunyi adalah 8 sehingga tahapan uji coba hanya akan dilakukan disekitar angka 8.

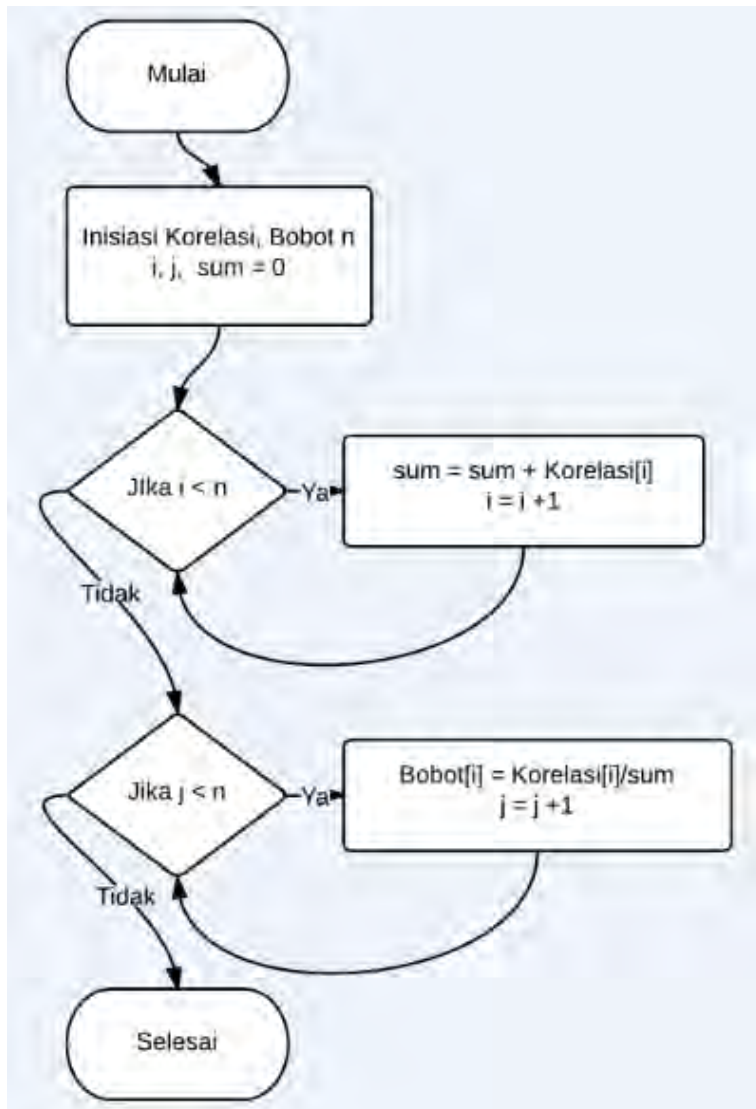
Proses inisiasi bobot dilakukan untuk ketiga jenis bobot dan dilakukan seperti yang ditunjukkan oleh diagram alur pada Gambar3.2.

Tahapan selanjutnya adalah melakukan proses pelatihan. Didalam proses pelatihan akan ada proses pemberian bobot jika hasil kelas prediksi tidak sama dengan hasil aktual. Proses pelatihannya secara detail dilakukan dengan langkah-langkah yang telah dijelaskan pada subbab 2.3.

3.4 Diagram Alir Jaringan Saraf Tiruan Regresi Linear

3.4.1 Inisiasi Bobot

Tidak seperti proses inisiasi bobot pada metode Jaringan Saraf Tiruan Multi Layer Perceptron, proses diinisiasi bobot pada metode Jaringan Saraf Tiruan Regresi Linear dilakukan dengan bantuan korelasi seperti yang telah dijelaskan sebelumnya pada Bab 2.3. Digram alir yang menggambarkan penjelasan inisiasi bobot pada subbab 2.3 dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram Alir Inisiasi Bobot Neural Netwrok Regresi Linear

3.4.2 Pelatihan

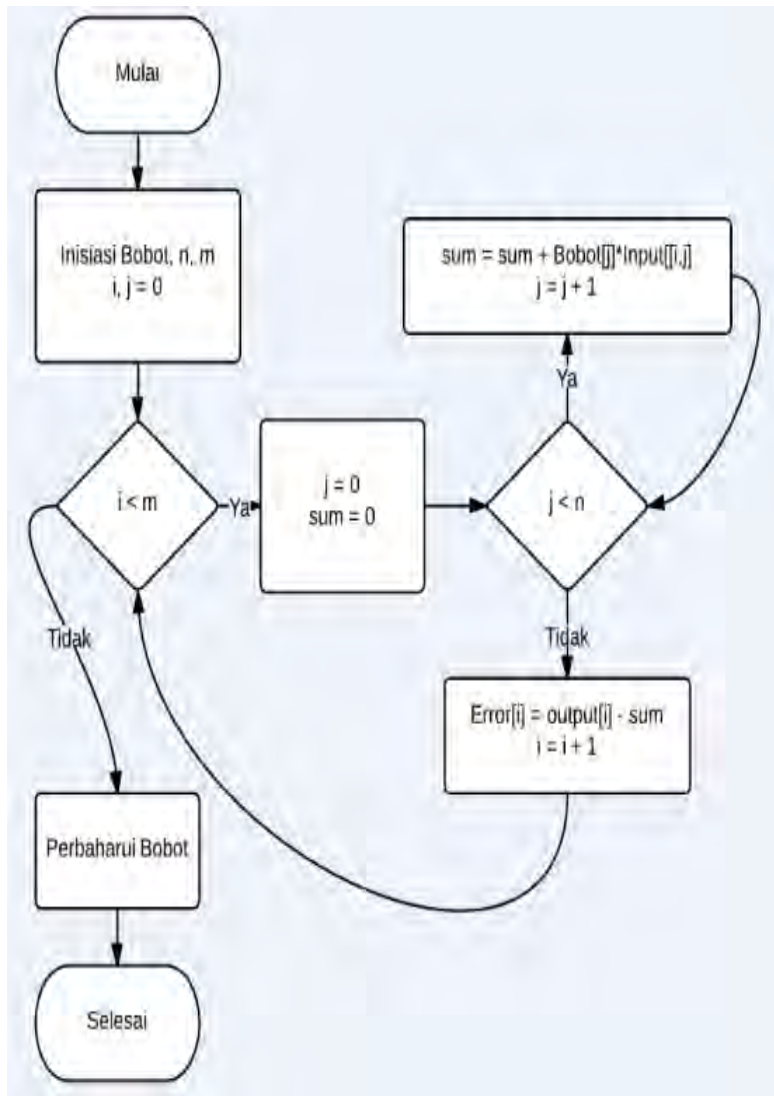
Secara umum Jaringan Saraf Tiruan Regresi Linear dalam Tugas Akhir ini dikerjakan dengan tahapan-tahapan seperti yang terlihat pada Gambar 3.5. Untuk setiap iterasinya dimulai dengan menentukan jumlah data yang akan diramalkan(m) dan jumlah data masukan(n). Pada iterasi pertama Bobot diinisiasi seperti pada Gambar 3.4 namun pada iterasi selanjutnya inisiasi Bobot diambil dari bobot yang telah diperbaharui pada iterasi sebelumnya.

Tahapan selanjutnya adalah melakukan iterasi sebanyak m dimana dalam setiap iterasi ini dilakukan perhitungan nilai ramalan untuk data ke i. Hasil ramalan didapatkan dengan menjumlahkan hasil perkalian antara bobot dengan data masukan. Nilai kesalahan untuk setiap data dihitung untuk proses pembaharuan bobot. Tidak seperti Jaringan Saraf Tiruan Multi Layer Perceptron, pada Jaringan Saraf Tiruan Regresi Linear hanya dilakukan satu kali pembaharuan bobot dalam satu kali iterasi tanpa memperdulikan jumlah data. Proses pembaharuan bobot dapat dilihat pada Bab 2.4.

3.5 Perancangan Antar Muka

Saat aplikasi dieksekusi pertama kali, pengguna akan berhadapan dengan sebuah form. Pada form tersebut akan ada tiga buah tab, yang pertama adalah “Inisiasi”, yang kedua adalah “Regresi”, dan yang ketiga adalah “MLP”.

Pada tab “Inisiasi” pengguna akan dihadapan dengan pilihan untuk mengatur kriteria data masukan dan iterasi masukan. Data masukan pertama sampai kelima dapat dipilih oleh pengguna dengan pilihan t-1, t-2, t-3, t-4, t-5, t-96, t-97, t-(96×7), t-(96*14), t-(96*21) dan t-(96*28). Penjelasan secara detail tentang masing-masing data masukan dapat dilihat pada subbab 2.2.1. Data masukan yang keenam hanya bisa dimasukan “Ya” atau “Tidak” karena data masukan keenam mengindikasikan pengaruh hari libur. Gambar untuk tab ini dapat dilihat pada Gambar 3.6.



Gambar 3.5 Diagram Alir Jaringan Saraf Tiruan Regresi Linear

Forecasting Traffic

Inisiasi Bobot | Regresi | MLP

Data Masukan Pertama :

Data Masukan Kedua :

Data Masukan Ketiga :

Data Masukan Keempat :

Data Masukan Kelima :

Jenis Hari Berpengaruh :

Iterasi Maksimum :

Gambar 3.6 Tampilan awal pada tab Inisiasi

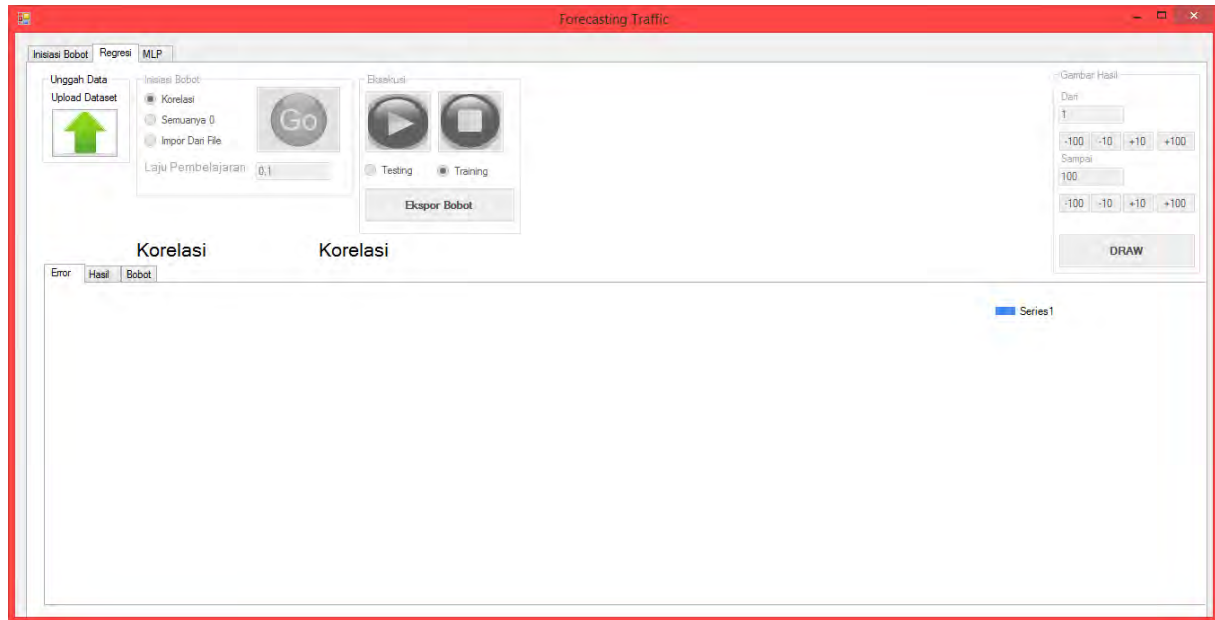
Pada tab selanjutnya, tab “Regresi”, pengguna akan dihadapkan dengan empat buah groupbox dan sebuah tabpages. Groupbox pertama akan berisi sebuah tombol yang digunakan mengunggah data masukan.

Selanjutnya adalah groupbox yang didalamnya terdapat beberapa radiobutton yang berguna untuk membantu pengguna memilih cara untuk menginisai bobot. Nilai laju pembelajaran juga diatur pada groupbox ini. Setelah selesai mengatur laju pembelajaran dan cara penginisiasian bobot, pengguna dapat menekan sebuah tombol “Go” untuk melanjutkan pada tahapan selanjutnya.

Groupbox yang ketiga berguna bagi pengguna untuk menentukan apakah yang ingin dicoba adalah proses pelatihan atau pengetestan. Terdapat juga dua tombol dalam groupbox ini yang berguna untuk memberhentikan proses dan menjalankannya kembali. Groupbox yang terakhir atau keempat berguna untuk menggambarkan hasil prediksi. Pengguna dapat mengatur data dari seberapa sampai seberapa yang ingin ditampilkan. Misalnya pengguna memilih ingin melihat hasil prediksi pada data kesatu sampai ke 100 maka akan tampil grafik yang menggambarkan hasil prediksi dan hasil aktual.

Seperti yang telah disebutkan sebelumnya, selain 4 buah groupbox pada tab “Regresi” juga terdapat sebuah tabpages yang terdiri dari 3 buah tab. Tab pertama berisi grafik yang menampilkan nilai kesalahan. Pengguna dapat melihat pergerakan nilai kesalahan mulai dari iterasi pertama sampai dengan iterasi terakhir pada tab ini. Pada tab selanjutnya terdapat sebuah grafik yang menggambarkan perbandingan antara hasil sebenarnya dan hasil prediksi. Pada tab terakhir pengguna dapat melihat pergerakan bobot mulai dari iterasi pertama sampai iterasi terakhir. Gambar untuk tab ini dapat dilihat pada Gambar 3.7.

Pada tab terakhir di form ini, yaitu tab “MLP”, pengguna akan dihadapkan dengan tampilan yang mirip dengan tampilan pada tab “Regresi”. Secara umum kegunaan dari semua groupbox dan tabpages di tab ini sama dengan kegunaannya di tab “Regresi”.



Gambar 3.7 Tampilan awal tab Regresi

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas tentang implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada Bab III.

4.1 Lingkungan Implementasi

Lingkungan implementasi mencakup penggunaan sumberdaya yang dipakai untuk menjalankan uji coba serta implementasi. Sumberdaya ini berupa perangkat keras dan perangkat lunak.

4.2 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan pada lingkungan pengembangan aplikasi adalah komputer PC dengan spesifikasi sebagai berikut :

- Prosesor yang digunakan Intel(R) Core(TM) i5 2430 CPU @ 2.40GHz(4 CPU)
- Memori yang dipakai RAM 6,00 GB
- Tipe sistem operasi 64-bit

4.3 Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

- Sistem operasi menggunakan *Windows 8.1 x64*
- Visual Studio sebagai IDE untuk mengimplementasikan aplikasi kedua metode Jaringan Saraf Tiruan Multi Layer Perceptron dan Jaringan Saraf Tiruan Regresi Linear
- C# sebagai bahasa pemrograman untuk mengimplementasikan metode Jaringan Saraf Tiruan Multi Layer Perceptron dan Jaringan Saraf Tiruan Regresi Linear.

4.4 Implementasi Kode Perangkat Lunak

Subbab ini akan dijelaskan mengenai implementasi proses perangkat lunak yang sudah dirancang pada Bab 3.

4.4.1 Implementasi Pembangkitan Data

Data yang akan dibangkitkan terdiri dari 3 buah distribusi, normal, *uniform* dan eksponensial. Tabel 4.1 dan Tabel 4.2 menggambarkan persebaran data yang akan dibangkitkan untuk menjadi data uji coba. Tabel 4.1 adalah jenis distribusi kendaraan pada hari aktif dan Tabel 4.2 berisi jenis distribusi kendaraan di hari libur. Semua data yang telah diambil dari jalan raya dapat dilihat pada lampiran A.

Tabel 4.1 dan 4.2 berisi lima buah kolom. Kolom pertama adalah rentang waktu awal dan kolom kedua adalah rentang waktu akhir. Kolom ketiga adalah jenis distribusi jumlah kendaraan pada rentang waktu yang telah ditentukan. Jika pada suatu baris kolom ini berisi “Normal” maka kolom keempat (Faktor1) adalah rata-rata dan kolom kelima (Faktor2) adalah standar deviasi namun jika berisi “Uniform” maka baris keempat (Faktor1) adalah jumlah minimum dan kolom kelima (Faktor2) adalah jumlah maksimum dari jumlah kendaraan. Jika pada kolom berisi “Eksponensial” maka pada kolom keempat (Faktor1) akan berisi rata-rata waktu kedatangan(dalam detik) dari distribusi tersebut dan kolom kelima (Faktor2) akan berisi “-“ yang tidak bermakna.

Pada Tabel 4.1 berisi distribusi jumlah kendaraan yang melewati suatu jalan pada hari libur. Dapat diartikan pada hari libur, jumlah kendaraan yang melewati suatu jalan pada jam 7 sampai 8 terdistribusi secara normal dengan rata-rata 8,24 dan standar deviasi 1,23. Pada jam 8 sampai 9, jumlah motor yang melewati jalan tersebut terdistribusi secara uniform dengan jumlah minimum 8 dan jumlah maksimum 11. Antara jam 9 sampai jam 10 jumlah kendaraan terdistribusi secara eksponensial dengan rata-rata waktu kedatangan berjarak 6,81

Tabel 4.1 Penyebaran Jumlah Kendaraan di Hari Libur

No	Mulai	Akhir	Distribusi	Faktor1	Faktor2
1	00.00	00.59	Uniform	0	5
2	01.00	01.59	Uniform	0	5
3	02.00	02.59	Uniform	0	5
4	03.00	03.59	Uniform	0	5
5	04.00	04.59	Uniform	0	5
6	05.00	05.59	Uniform	0	5
7	06.00	06.59	Ekspensial	7,29	-
8	07.00	07.59	Normal	8,24	1,23
9	08.00	08.59	Uniform	8	11
10	09.00	09.59	Ekspensial	6,81	-
11	10.00	10.59	Normal	10,51	1,65
12	11.00	11.59	Ekspensial	5,10	-
13	12.00	12.59	Normal	12,21	3,67
14	13.00	13.59	Uniform	13	16
15	14.00	14.59	Uniform	13	16
16	15.00	15.59	Ekspensial	6,89	-
17	16.00	16.59	Ekspensial	7,12	-
18	17.00	17.59	Uniform	8	12
19	18.00	18.59	Ekspensial	5,97	-
20	19.00	19.59	Uniform	10	14
21	20.00	20.59	Uniform	8	12
22	21.00	21.59	Uniform	6	8
23	22.00	22.59	Uniform	5	7
24	23.00	23.59	Uniform	5	7

Seperti yang telah dibahas sebelumnya, dalam Tugas akhir ini hanya ada tiga jenis pemodelan distribusi untuk jumlah kendaraan yaitu normal, *uniform* dan ekspensial, sehingga Pada Tabel 4.1, Tabel 4.2, Tabel 4.3 dan Tabel 4.4 yang menggambarkan pola penyebaran kendaraan pada suatu jalan tertentu hanya mengikuti ketiga distribusi tersebut.

Tabel 4.2 Penyebaran Jumlah Kendaraan di Hari Aktif

Mulai	Akhir	Distribusi	Faktor1	Faktor2
00.00	00.59	Uniform	0	5
01.00	01.59	Uniform	0	5
02.00	02.59	Uniform	0	5
03.00	03.59	Uniform	0	5
04.00	04.59	Uniform	0	5
05.00	05.59	Uniform	3	8
06.00	06.59	Ekspensial	4,45	-
07.00	07.59	Normal	10,41	0,97
08.00	08.59	Uniform	7	10
09.00	09.59	Uniform	6	10
10.00	10.59	Uniform	7	10
11.00	11.59	Uniform	12	14
12.00	12.59	Ekspensial	3,39	-
13.00	13.59	Uniform	12	14
14.00	14.59	Uniform	13	16
15.00	15.59	Uniform	13	16
16.00	16.59	Uniform	12	14
17.00	17.59	Ekspensial	4,65	-
18.00	18.59	Ekspensial	6,32	-
19.00	19.59	Uniform	10	14
20.00	20.59	Uniform	8	12
21.00	21.59	Uniform	6	8
22.00	22.59	Uniform	5	7
23.00	23.59	Uniform	5	7

Implementasi kode untuk membangkitkan data yang terdistribusi secara normal dapat dilihat pada Kode Sumber 4.1, implementasi kode untuk membangkitkan data yang terdistribusi secara *uniform* dapat dilihat pada Kode Sumber 4.2 dan implementasi kode untuk membangkitkan data yang terdistribusi secara eksponensial dapat dilihat pada Kode Sumber 4.3. Semua simbol yang digunakan pada ketiga Kode Sumber tersebut dapat dilihat pada Tabel 3.1.

```

Random rand = new Random();
double u1 = rand.NextDouble();
double u2 = rand.NextDouble();
double randStdNormal = Math.Sqrt(-2.0 *
Math.Log(u1)) * Math.Sin(2.0 * Math.PI * u2);
double randNormal = mean + stdDev * randStdNormal;

```

Kode Sumber 4.1 Implementasi Kode Untuk Membangkitkan Data dengan Distribusi Normal

```

Random rand = new Random();
double u = rand.NextDouble();
double randUniform = a + (b-a)*u

```

Kode Sumber 4.2 Implementasi Kode Untuk Membangkitkan Data dengan Distribusi Uniform

```

Random rand = new Random();
double u = rand.NextDouble();
double randEksponensial = -1 * Math.Log(1-u) * mean

```

Kode Sumber 4.3 Implementasi Kode Untuk Membangkitkan Data dengan Distribusi Eksponensial

4.4.2 Implementasi Jaringan Saraf Tiruan Multi Layer Perceptron

Banyak proses yang akan dilalui saat akan mengimplementasi Jaringan Saraf Tiruan Multi Layer Perceptron seperti yang telah dijelaskan di subbab 2.3. Proses-proses tersebut adalah :

1. Memilih Data masukan

Kode yang digunakan untuk mengimplementasi tahapan ini dapat dilihat pada Kode Sumber 4.4. Fungsi “MakeInput”

mempunyai parameter fungsi matrix dan i1 hingga i5. Parameter matrix berisi data masukan secara keseluruhan. Parameter i1 hingga i5 berisi kriteria input yang diinginkan. Seperti yang telah dijelaskan, dalam tahapan uji coba akan dilakukan banyak jenis kombinasi data masukan untuk mencari kombinasi yang terbaik. Karena data jumlah kendaraan yang diobservasi adalah setiap 15 menit, maka akan ada 96 kelompok dalam setiap harinya. Jenis data masukan yang akan diuji coba berpeluang sampai pada 1 bulan sebelumnya oleh karena itu dipilih angka 30(1 bulan 30 hari) sebagai batas minimum iterasi.

2. Menentukan Jumlah Iterasi Maksimum

Jumlah iterasi maksimum diimplementasikan seperti pada Kode Sumber 4.5. iterasimaksimum adalah variabel yang digunakan untuk mencatat masukan dari pengguna. IterMaksTxt adalah nama dari TextButton pada tampilan antarmuka.

3. Inisiasi Bobot

Pada Kode Sumber 4.6 akan dijabarkan proses inisiasi bobot. Dapat dilihat pada fungsi “InitialWeight” terdapat empat buah parameter yaitu min, max, rows dan columns. Parameter min menandakan batas bawah yang diperbolehkan, max adalah batas atas yang diperbolehkan, rows dan columns adalah dimensi matriks bobot. Seperti yang telah dijelaskan sebelumnya, setiap bobot terdiri dari 2 dimensi dan harus berada pada rentang -1 sampai 1.

```
private void SetIterasiMaksBtn_Click(object sender,
EventArgs e)
{
    iterasimaksimum=Convert.ToInt32(IterMaksTxt.Text);
}
```

Kode Sumber 4.4 Implementasi iterasi maksimal

```

public static double[][] MakeInput(Matrix matrix,
int i1, int i2, int i3, int i4, int i5)
{
    int n = matrix.GetData()[0].GetLength(0);
    double[][] temp = new double[Layout.jumlahInput][];
    for (int i = 0; i < Layout.jumlahInput; i++)
    {
        temp[i] = new double[n - 96 * 30];
    }
    for (int i = 96 * 30; i < n; i++)
    {
        temp[0][i - 96 * 30] = matrix.GetData()[0][i1];
        temp[1][i - 96 * 30] = matrix.GetData()[0][i2];
        temp[2][i - 96 * 30] = matrix.GetData()[0][i3];
        temp[3][i - 96 * 30] = matrix.GetData()[0][i4];
        temp[4][i - 96 * 30] = matrix.GetData()[0][i5];
    }
    return temp;
}

```

Kode Sumber 4.5 Implementasi fungsi makeinput

```

public static Matrix InitialWeight(double min,
double max, int rows, int columns)
{
    Matrix W = new Matrix(rows, columns);
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            W.data[i][j] = Randoming(min, max);
        }
    }
    return W;
}

```

Kode Sumber 4.6 Implementasi fungsi initialweight

```

private static double Randoming(double up, double
down)
{
    double R = new Random().NextDouble();
    System.Threading.Thread.Sleep(10);
    return (down += (up - down) * R);
}

```

Kode Sumber 4.7 Implementasi fungsi randoming

4. Pelatihan

Proses pelatihan meliputi proses klasifikasi, mencari nilai kesalahan dan memperbaiki bobot. Implementasi dari ketiga tahapan diatas dalam kode dapat dilihat pada Kode Sumber 4.8. Proses klasifikasi dikerjakan dengan Kode Sumber 4.9, proses menghitung nilai kesalahan dengan Kode Sumber 4.10 dan proses memperbaiki bobot dengan Kode Sumber 4.11.

Kode Sumber 4.8 adalah proses pelatihan secara keseluruhan. Proses pelatihan, seperti yang telah dijelaskan di subbab 2.4, mempunyai 3 buah tahapan secara umum. Tahapan pertama adalah klasifikasi yang diimplementasikan dengan Kode Sumber 4.9. Tahapan ini mengimplementasikan Persamaan 2.11 dan Persamaan 2.12.

Kode Sumber 4.10 adalah proses implementasi dari Persamaan 2.13 dan Persamaan 2.14 yang berguna untuk menghitung nilai kesalahan pada lapisan tersembunyi dan lapisan keluaran. Kode Sumber 4.11 adalah pengimplementasian dari cara pembaruan bobot pada subbab 2.4.


```

public static int Training(Matrix V, Matrix W, Matrix
Bias)
{
    int error = 0;
    int n = V.GetData().GetLength(0);
    int m =
    MultiLayerPerceptron.inputNormal.GetData()[0].GetL
    ength(0);
    double[] outputHidden = new double[neuronHidden];
    double[] output = new double[jumlahKelas];
    double[] errorHidden = new double[neuronHidden];
    double[] errorOutput = new double[jumlahKelas];
    for (int i = 0; i < m; i++)
    {
        #region Klasifikasi
        if (MultiLayerPerceptron.outputPredict.data[0][i]
        !=
        MultiLayerPerceptron.outputNormal.GetData()[0][i])
        {
            error++;
            #region HitungErrorOutputdanHidden
            #region UpdateBobotDanBias
        }
    }
    return error;
}

```

Kode Sumber 4.8 Implementasi fungsi pelatihan Jaringan Saraf Tiruan Multi Layer Perceptron

Pada Kode Sumber 4.8 terdapat 3 buah region yang menggambarkan 3 buah proses utama dalam proses pelatihan, yaitu region Klasifikasi, region HitungErrorOutputdanHidden dan region UpdateBobotDanBias.

```

#region Klasifikasi
for (int j = 0; j < neuronHidden; j++)
{
    outputHidden[j] = 0;
    for (int k = 0; k < n; k++)
    {
        outputHidden[j] +=
            (V.GetData()[k][j]*inputNormal.GetData()[k][i]);
    }
    outputHidden[j] += Bias.GetData()[0][j];
    outputHidden[j] = 1 / (1 +
        Math.Exp(outputHidden[j] * -1));
    }
    for (int j = 0; j < jumlahKelas; j++)
    {
        output[j] = 0;
        for (int k = 0; k < neuronHidden; k++)
        {
            output[j] += (W.GetData()[k][j] *
                outputHidden[j]);
        }
        output[j] += Bias.GetData()[1][j];
        output[j] = 1 / (1 + Math.Exp(output[j] * -1));
    }
    double max = -100;
    for (int j = jumlahKelas - 1; j >= 0; j--)
    {
        if (output[j] > max)
        {
            max = output[j];
            MultiLayerPerceptron.outputPredict.data[0][i] = j;
        }
    }
}
#endregion

```

Kode Sumber 4.9 Implementasi #region klasifikasi pada Kode Sumber 4.8

```

#region HitungErrorOutputDanHidden
for (int j = 0; j < jumlahKelas; j++)
{
    double temp = 0;
    if
    (MultiLayerPerceptron.outputNormal.GetData()[0][
    i] == j)
    {
        temp = 1;
    }
    errorOutput[j] = output[j] * (1 - output[j]) *
    (temp - output[j]);
    for (int k = 0; k < neuronHidden; k++)
    {
        errorHidden[j] = outputHidden[j] * (1 -
        outputHidden[j]);
        double sum = 0;
        for (int k = 0; k < jumlahKelas; k++)
        {
            sum += (W.GetData()[j][k] *
            errorOutput[k]);
        }
        errorHidden[j] *= sum;
    }
}
#endregion

```

***Kode Sumber 4.10 Implementasi #region
hitungErrorOutputDanHidden pada Kode Sumber 4.8***

```

#region UpdateBobotDanBias
for (int j = 0; j < jumlahKelas; j++)
{
    for (int k = 0; k < neuronHidden; k++)
    {
        W.data[k][j] += (learningW * errorOutput[j] *
            outputHidden[k]);
    }
    Bias.data[1][j] += (learningW * errorOutput[j]);
}
for (int j = 0; j < neuronHidden; j++)
{
    for (int k = 0; k < n; k++)
    {
        V.data[k][j] += (learningW * errorHidden[j] *
            inputNormal.GetData()[k][i]);
    }
    Bias.data[0][j] += (learningW * errorHidden[j]);
    //MessageBox.Show((learningW *
        errorHidden[j]).ToString());
}
}
#endregion

```

Kode Sumber 4.11 Implementasi #region updateBobosDanBias pada Kode Sumber 4.8

4.4.3 Implementasi Jaringan Saraf Tiruan Regresi Linear

Implementasi dalam bentuk kode untuk metode Jaringan Saraf Tiruan Regresi Linear dengan tahapan yang sesuai dengan yang telah dijelaskan pada subbab 2.5 akan dijelaskan pada bagian ini.

1. Memilih Data Masukan dan Menentukan Iterasi Maksimum

Tahapan pengerjaan ini dikerjakan dengan cara yang sama seperti pada metode Jaringan Saraf Tiruan Multi

Layer Perceptron yaitu dengan memanfaatkan Kode Sumber 4.1 dan Kode Sumber 4.2.

2. Inisiasi Bobot

Seperti yang telah dijelaskan pada subbab 2.5, untuk melakukan inisiasi bobot pada metode Jaringan Saraf Tiruan Regresi Linear dibutuhkan terlebih dahulu perhitungan terhadap korelasi antara data masukan yang terkait dengan data keluaran. Implementasi kodenya dapat dilihat pada Kode Sumber 4.12.

```
public static double Corelation(double[] A, double[]
B)
{
    double mean1 = GetMean(A);
    double mean2 = GetMean(B);
    int n = A.GetLength(0);
    double sum1 = 0;
    double sum2 = 0;
    double sum3 = 0;
    for (int i = 0; i < n; i++)
    {
        sum1 += ((A[i] - mean1) * (B[i] - mean2));
        sum2 += ((A[i] - mean1) * (A[i] - mean1));
        sum3 += ((B[i] - mean2) * (B[i] - mean2));
    }
    return (double) (sum1/Math.Sqrt(sum2*sum3));
}
```

Kode Sumber 4.12 Implementasi penghitungan korelasi

Setelah mendapatkan nilai korelasi antara masing-masing data masukan dengan data keluaran maka dapat dihitung bobot untuk masing-masing data masukan. Bobot-bobot tersebut diberikan kepada masing-masing data masukan berdasarkan presentasi dari korelasinya.

Implementasi untuk inisiasi bobot tersebut dapat dilihat pada Kode Sumber 4.13.

```
public static Matrix Normalisasi (Matrix A)
{
    int n = A.GetData().GetLength(0);
    int m = A.GetData()[0].GetLength(0);
    double sum = 0;
    for (int i = 0; i < 1; i++)
    {
        for (int j = 0; j < m; j++)
        {
            sum += A.GetData()[i][j];
        }
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            A.data[i][j] = A.GetData()[i][j]/sum;
        }
    }
    return A;
}
```

Kode Sumber 4.13 Implementasi normalisasi korelasi

3. Pelatihan

Proses pelatihan meliputi proses prediksi, mencari nilai kesalahan dan memperbaiki bobot. Implementasi dari ketiga tahapan diatas dalam kode dapat dilihat pada Kode Sumber 4.14, Kode Sumber 4.15 dan Kode Sumber 4.16. Pada Kode Sumber 4.14 terdapat 2 buah region, yang diimplementasikan pada Kode Sumber 4.15 dan 4.16, yang berisi dua buah proses utama yaitu prediksi dan perhitungan nilai kesalahan.

```

public static double Train(Matrix W)
{
    int n = inputNormal.GetData().GetLength(0);
    int m = inputNormal.GetData()[0].GetLength(0);
    double error = 0;
    double[] sum = new double[m];
    #region prediksiDanMenghitungError
    #region updateBobot
    return error / m;
}

```

Kode Sumber 4.14 Implementasi fungsi pelatihan Jaringan Saraf Tiruan Regresi Linear

```

#region prediksiDanMenghitungError
for (int i = 0; i < m; i++)
{
    sum[i] = 0;
    for (int j = 0; j < n; j++)
    {
        sum[i] += (W.GetData()[0][j] *
            inputNormal.GetData()[j][i]);
    }
    Layout.outputPredict.GetData()[0][i] = (sum[i]);
    error +=
        Math.Abs((int)(outputNormal.GetData()[0][i] *
            inputMaximum) - (int)(sum[i] * inputMaximum));
}
#endregion

```

Kode Sumber 4.15 Implementasi #region prediksiDanMenghitungError pada Kode Sumber 4.14

```

#region UpdateBobotDanBias
for (int j = 0; j < n; j++)
{
    double temp =0;
    for (int i = 0; i < m; i++)
    {
        temp += (outputNormal.GetData()[0][i] - sum[i]) *
        inputNormal.GetData()[j][i];
    }
    if (j == 5) temp *= -2;
    else temp *= -2 * learningRate;
    W.data[0][j] -= (temp/m);
}
#endregion

```

Kode Sumber 4.16 Implementasi #region UpdateBobotDanBias pada Kode Sumber 4.14

4.5 Implementasi Antarmuka

4.5.1 Antarmuka Pembangkitan Data Ujicoba

Gambar 4.1 adalah tampilan yang akan dihadapkan pada pengguna saat akan melakukan pembangkitan data uji coba. Data yang dibangkitkan mengikuti kriteria pada Tabel 4.1 dan Tabel 4.2. Pada antarmuka tersebut pengguna akan dihadapkan pada banyak pilihan rentang waktu, mulai dari rentang 00.00 sampai 00.59 hingga 23.00 sampai 23.59. Untuk setiap rentang waktunya, pengguna dihadapkan pada sebuah combobox dan dua buah textbox. Pada setiap combobox pengguna dapat memilih 3 pilihan, yaitu “normal”, “uniform” dan “eksponensial”. Jika yang dipilih pada combobox adalah “normal” maka textbox pertama akan menunjukkan rata-rata dan textbox yang kedua akan menunjukkan standar deviasi. Jika yang dipilih adalah “uniform” maka textbox yang pertama adalah batas bawah dan textbox yang kedua adalah batas atas.

Form Pembangkitan Jumlah Kendaraan

Bangkitkan

Distribusi Jumlah Kendaraan Berdasarkan Waktu

Hari Libur ▼

00.00 - 01.00	Uniform ▼	0	5	12.00 - 13.00	Uniform ▼	0	5
01.00 - 02.00	Uniform ▼	0	5	13.00 - 14.00	Uniform ▼	0	5
02.00 - 03.00	Uniform ▼	0	5	14.00 - 15.00	Uniform ▼	0	5
03.00 - 04.00	Uniform ▼	0	5	15.00 - 16.00	Uniform ▼	0	5
04.00 - 05.00	Uniform ▼	0	5	16.00 - 17.00	Uniform ▼	0	5
05.00 - 06.00	Uniform ▼	0	5	17.00 - 18.00	Uniform ▼	0	5
06.00 - 07.00	Uniform ▼	0	5	18.00 - 19.00	Uniform ▼	0	5
07.00 - 08.00	Uniform ▼	0	5	19.00 - 20.00	Uniform ▼	0	5
08.00 - 09.00	Uniform ▼	0	5	20.00 - 21.00	Uniform ▼	0	5
10.00 - 11.00	Uniform ▼	0	5	22.00 - 23.00	Uniform ▼	0	5
11.00 - 12.00	Uniform ▼	0	5	23.00 - 24.00	Uniform ▼	0	5

Simpan

Gambar 4.1 Tampilan Antarmuka Halaman Pembangkitan Data Uji Coba

4.5.2 Antarmuka Pemilihan Data Masukan dan Iterasi Maksimum

Gambar 4.2 menampilkan cara pengaturan data masukan dan iterasi maksimum, contoh kombinasi data masukan pada gambar tersebut sama dengan contoh kombinasi data masukan pada subbab 2.2.1.

The interface is titled 'Inisiasi Bobot' and includes three tabs: 'Inisiasi Bobot', 'Regresi', and 'MLP'. The 'Inisiasi Bobot' tab is active. Below the tabs, there are six dropdown menus for selecting input data:

- Data Masukan Pertama : t-1
- Data Masukan Kedua : t-2
- Data Masukan Ketiga : t-96
- Data Masukan Keempat : t-97
- Data Masukan Kelima : t-96*7
- Jenis Hari Berpengaruh : Ya

Below these dropdowns, there is a text input field for 'Iterasi Maksimum' with the value '15000'. To the right of this field is a button labeled 'Atur'.

Gambar 4.2 Antarmuka pemilihan data masukan dan iterasi maksimum

4.5.3 Antarmuka Jaringan Saraf Tiruan Multi Layer Perceptron

Berbagai tahapan yang harus dilewati oleh pengguna untuk dapat menjalankan metode ini adalah :

1. Menggunggah Data

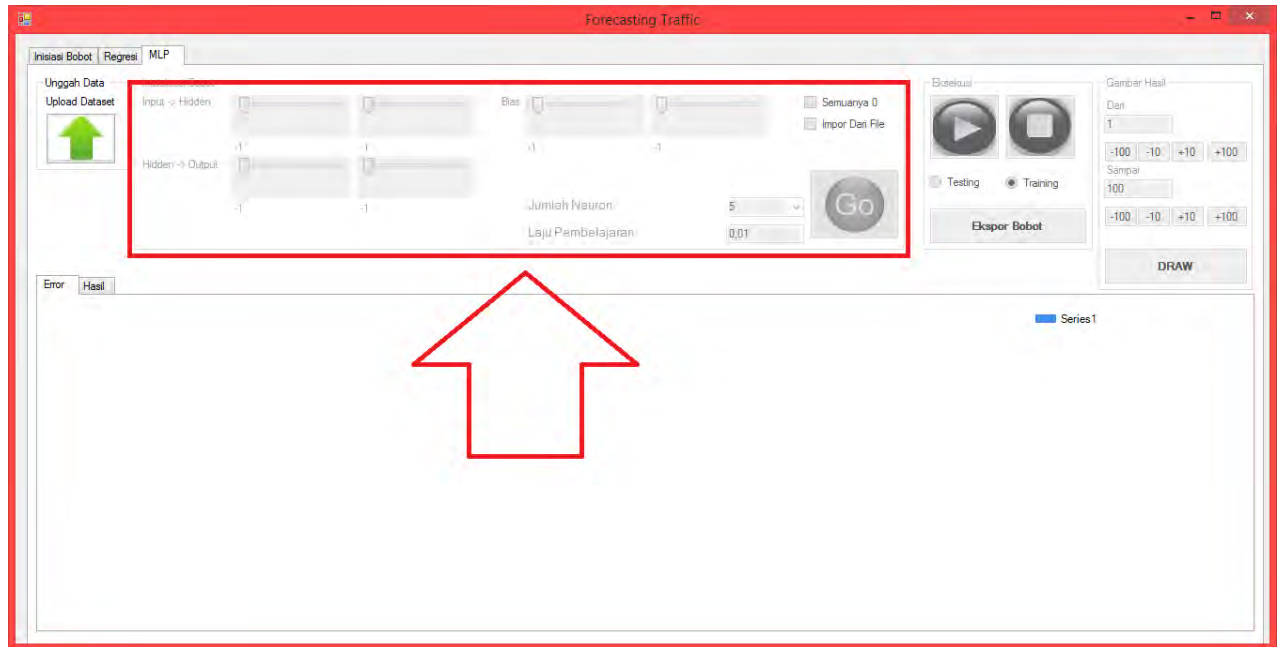
Tahapan awal adalah proses menggunggah data, hal ini dapat dilakukan dengan menekan tombol seperti pada Gambar 4.2.

2. Inisiasi Bobot

Ada tiga jenis bobot yang diinisiasi pada tahapan inisiasi bobot yaitu bobot dari lapisan masukan sampai lapisan tersembunyi, bobot dari lapisan tersembunyi sampai lapisan keluaran dan nilai bias yang proses penginisiasinya masing-masing ditunjukkan oleh angka 1, 2 dan 3 pada Gambar 4.4. Tahapan penting lainnya yang harus diinisiasi sebelum menjalankan program adalah menentukan jumlah *neuron* pada lapisan tersembunyi dan nilai laju pembelajaran dimana ditunjukkan nomor 4 pada Gambar 4.4. Setelah semua proses selesai maka pengguna dapat menekan tombol Go untuk meneruskan ke tahapan selanjutnya.

3. Hasil

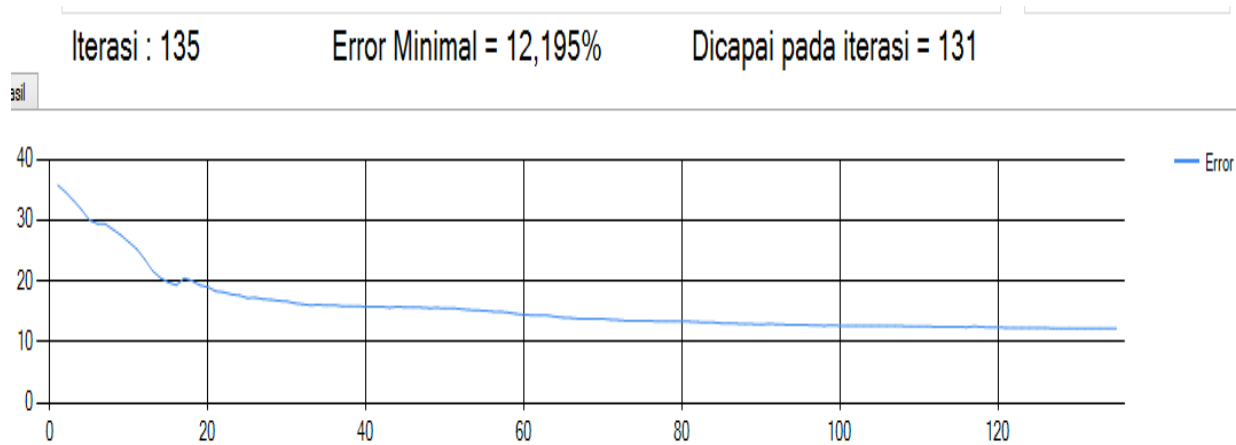
Pada Gambar 4.5, memperlihatkan laju perubahan nilai kesalahan pada 135 iterasi pertama. Dapat dilihat bahwa dalam 135 iterasi pertama nilai kesalahan minimal yang dapat diperoleh adalah 12,195% dan nilai kesalahan itu terjadi pada iterasi ke-131. Pada Gambar 4.6, dapat dilihat hasil peramalan yang telah dihitung dalam 135 iterasi pertama. Ada dua buah diagram yang dipertujukan yaitu yang berwarna kuning dan biru. Diagram garis biru yang tertutup dengan diagram kuning menandakan bahwa hasil ramalan pada data tersebut telah benar, namun diagram biru yang belum tertutup oleh diagram garis kuning berarti pada data tersebut masih ada kesalahan peramalan.



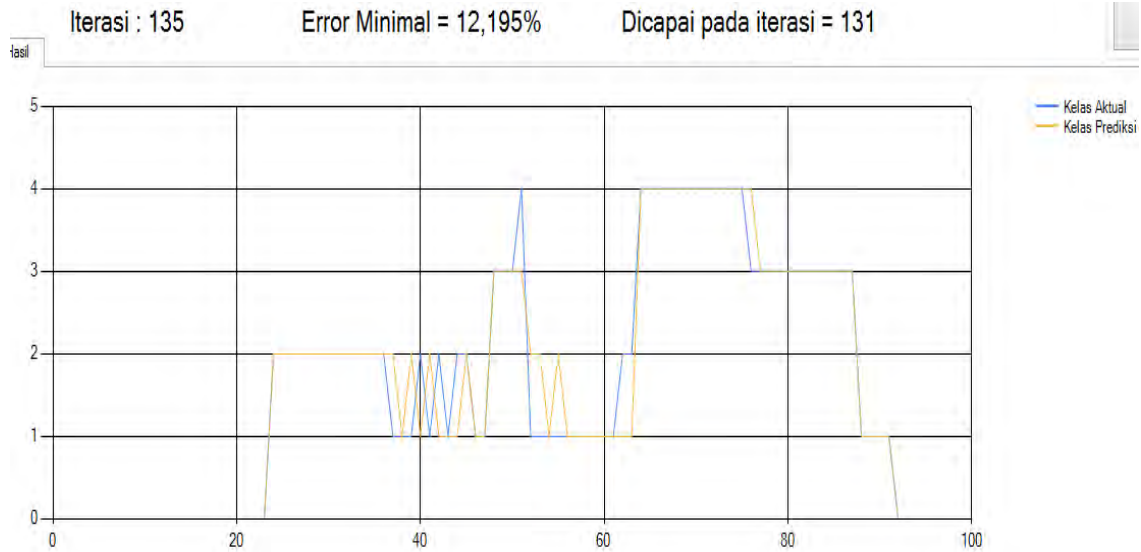
Gambar 4.3 Bagian inisiasi bobot Jaringan Saraf Tiruan Multi Layer Perceptron



Gambar 4.4 Bagian detail inisiasi bobot Jaringan Saraf Tiruan Multi Layer Perceptron



Gambar 4.5 Tampilan nilai kesalahan Jaringan Saraf Tiruan Multi Layer Perceptron



Gambar 4.6 Hasil peramalan pada Jaringan Saraf Tiruan Multi Layer Perceptron

4.5.4 Antarmuka Jaringan Saraf Tiruan Multi Regresi Linear

Tampilan yang dapat dilihat pengguna pada perangkat lunak untuk metode ini dapat dilihat pada Gambar 4.7, Gambar 4.8 dan Gambar 4.10. Perbedaan pada ketiga gambar tersebut adalah masing-masing menunjukkan grafik yang berbeda dimana Gambar 4.7 menunjukkan grafik perubahan nilai kesalahan, Gambar 4.8 dan Gambar 4.9 menunjukkan grafik hasil prediksi sedangkan Gambar 4.10 menunjukkan grafik perubahan bobot. Untuk ketiga gambar tersebut data masukan yang digunakan adalah $t-1$, $t-2$, $t-96$, $t-97$ dan $t-(96 \times 7)$.

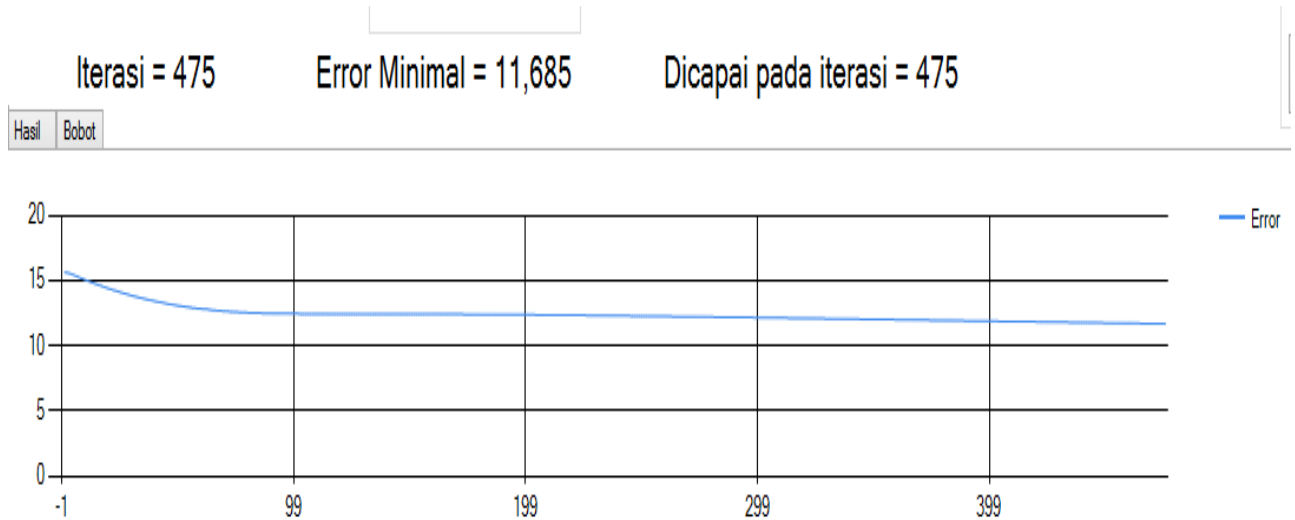
Dapat dilihat pada Gambar 4.7 bahwa terjadi perubahan nilai kesalahan yang cukup baik. Cepat atau lambatnya perubahan nilai kesalahan ditentukan oleh laju pembelajaran seperti yang telah dibahas sebelumnya. Jika nilai kesalahan pada Gambar 4.7 dibandingkan dengan nilai kesalahan pada Gambar 4.8 dapat disimpulkan bahwa nilai kesalahannya telah berkurang jauh berkurang jauh seiring dengan bertambahnya jumlah iterasi.

Gambar 4.8 dan 4.9 sama-sama menunjukkan hasil peramalan, hanya saja pada Gambar 4.8 hasil peramalan digambarkan saat nilai kesalahan masih berada pada angka 11,83 dan pada Gambar 4.9 hasil peramalan digambarkan saat nilai kesalahan berada pada angka 3,485. Perhitungan nilai kesalahan yang digunakan untuk metode Jaringan Saraf Tiruan Regresi Linear adalah *Mean Absolute Percentage Error*. Grafik garis yang berwarna kuning pada kedua gambar tersebut adalah nilai peramalan dan grafik garis biru adalah nilai aktual.

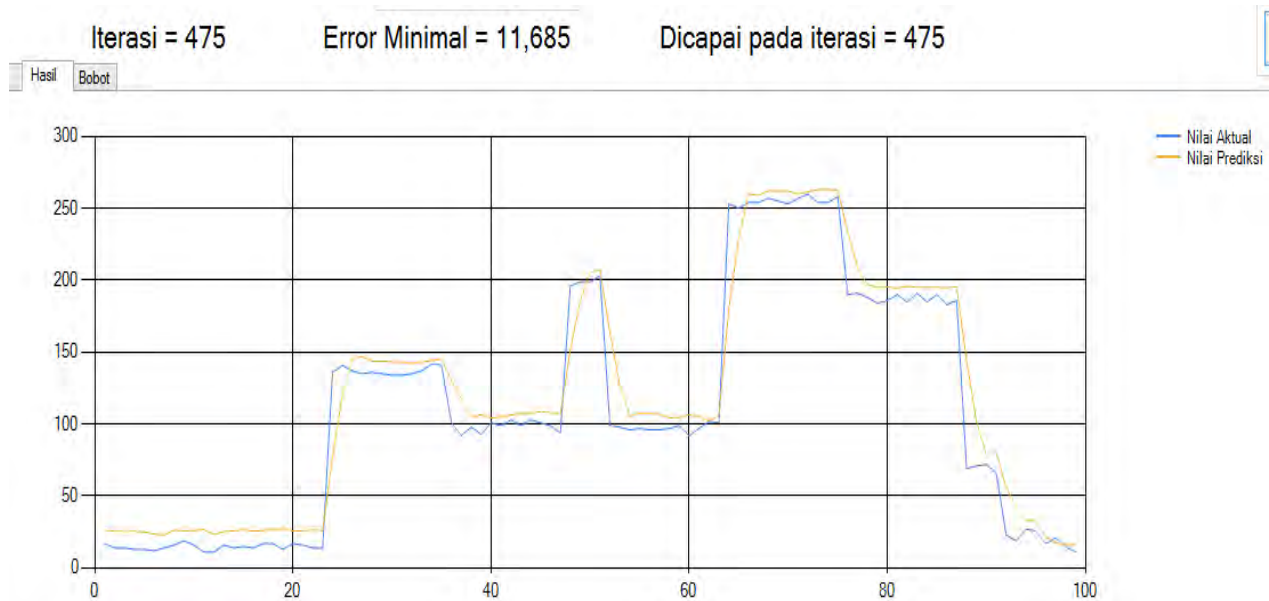
Dibandingkan dengan hasil peramalan pada Gambar 4.8, hasil peramalan pada Gambar 4.9 menghasilkan hasil peramalan yang jauh lebih akurat, hal ini dapat dilihat dengan grafik garis kuning lebih dekat pada grafik garis biru pada Gambar 4.9 dibandingkan Gambar 4.8.

Grafik garis ketiga yang dapat dilihat pada tampilan antarmuka adalah perubahan bobot dimana dapat dilihat pada Gambar 4.10. Pada percobaan ini diawal iterasi semua bobot

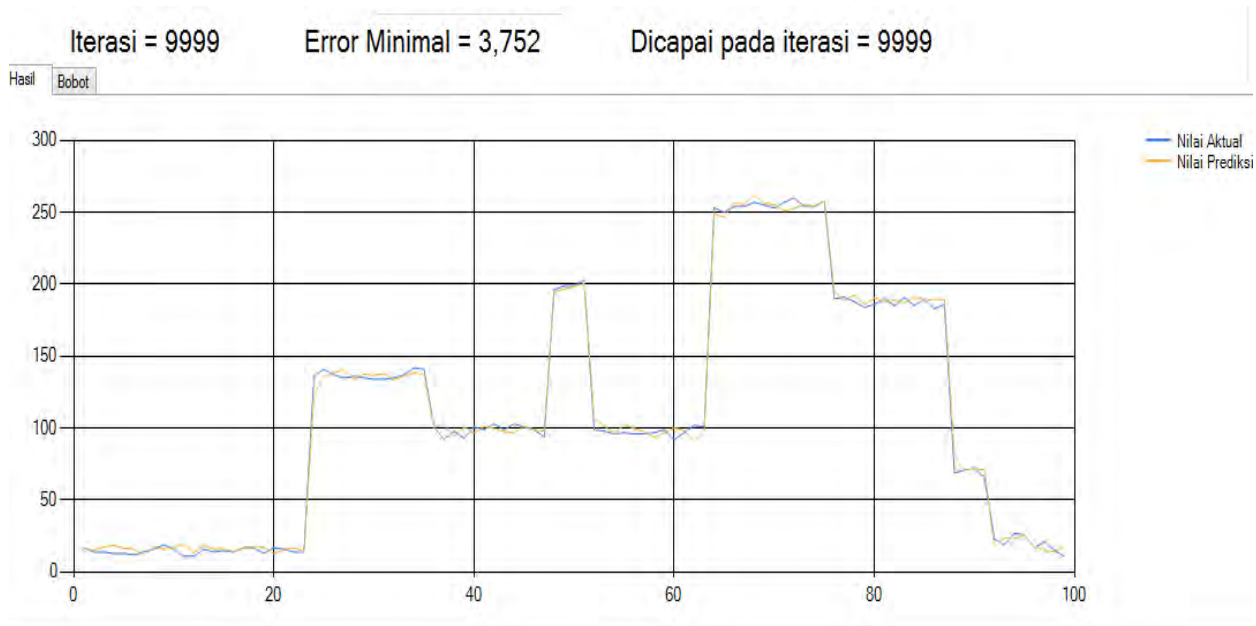
diinisiasi dengan bantuan korelasi. Dari Gambar 4.10 dapat dilihat bahwa terjadi perubahan yang signifikan pada bobot untuk data masukan keempat dan kelima dimana bobot untuk data masukan keempat terus menurun dan bobot untuk data masukan kelima terus meningkat.



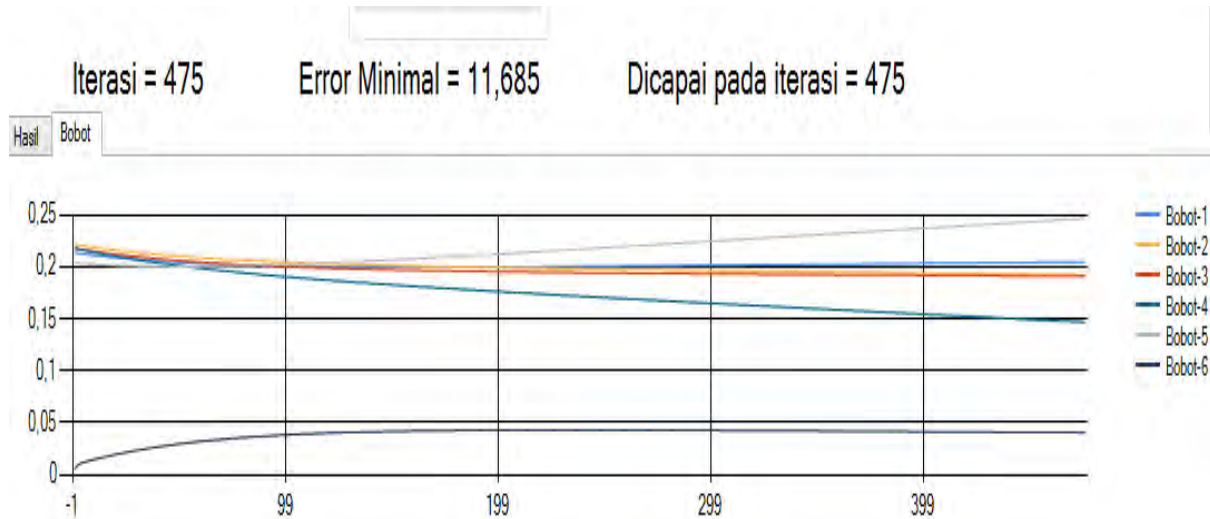
Gambar 4.7 Tampilan nilai kesalahan Jaringan Saraf Tiruan Regresi Linear



Gambar 4.8 Hasil Peramalan 1 Jaringan Saraf Tiruan Regresi Linear



Gambar 4.9 Hasil Peramalan 2 Jaringan Saraf Tiruan Regresi Linear



Gambar 4.10 Grafik bobot Neural Network Regresi Linear

BAB V

UJI COBA DAN EVALUASI

Pada bab ini dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan terhadap sistem yang dibuat. Pembahasan yang dipaparkan meliputi lingkungan uji coba, data uji coba, skenario uji coba, hasil uji coba, dan evaluasi.

5.1 Lingkungan Uji Coba

Lingkungan pengujian disini menjelaskan perangkat keras serta perangkat lunak yang digunakan dalam menjalankan beberapa skenario pengujian. Spesifikasi dari perangkat keras dan perangkat lunak ditunjukkan pada

Tabel 5.1 Spesifikasi Lingkungan Pengujian

Perangkat Keras	Prosesor	Intel(R) Core(TM) i5 2430 CPU @ 2.40GHz(4 CPU)
	Memori RAM	6.00 GB
Perangkat Lunak	Sistem Operasi	Windows 8.1 x64
	Perangkat	Visual Studio

5.2 Data Uji Coba

Data uji coba yang akan digunakan adalah data yang dibangkitkan dengan mengikuti pola-pola persebaran jumlah kendaraan seperti pada subbab 4.4.1.

5.3 Skenario Uji Coba

Pada subbab ini dijelaskan skenario uji coba yang dilakukan dan hasil yang didapatkan. Uji coba untuk Jaringan Saraf Tiruan Multi Layer Perceptron akan dilakukan untuk mencari kombinasi yang paling optimal untuk lapisan masukan, mencari jumlah neuron yang tepat pada lapisan tersembunyi dan mencari nilai laju

pembelajaran yang dapat menghasilkan akurasi paling tinggi. Karena pada Jaringan Saraf Tiruan Regresi Linear tidak ada lapisan tersembunyi, maka ujicoba pada metode ini hanya akan dilakukan pada lapisan masukan dan laju pembelajaran. Detail skenario uji coba dapat dilihat pada Tabel 5.2. Pada kolom “Metode” jika bernilai 1 artinya bahwa pada skenario tersebut diujikan pada metode Jaringan Saraf Tiruan dan 2 berarti skenario tersebut diujikan pada Jaringan Saraf Tiruan Regresi Linear.

Tabel 5.2 Skenario Uji Coba

Skenario	Metode	Tujuan
1	1	Mendapatkan kombinasi data masukan yang terbaik untuk metode Jaringan Saraf Tiruan Multi Layer Perceptron
2	1	Mendapatkan jumlah neuron paling optimal pada lapisan tersembunyi pada metode Jaringan Saraf Tiruan Multi Layer Perceptron
3	1	Mendapatkan nilai laju reaksi terbaik untuk metode Jaringan Saraf Tiruan Multi Layer Perceptron
4	2	Mendapatkan kombinasi data masukan yang terbaik untuk metode Jaringan Saraf Tiruan Regresi Linear
5	2	Mendapatkan nilai laju reaksi terbaik untuk metode Jaringan Saraf Tiruan Regresi Linear
6	1 dan 2	Mendapatkan akurasi pada data pengujian dengan menggunakan kombinasi terbaik dari kedua metode diatas

5.4 Pengujian dan Evaluasi

5.4.1 Lapisan Masukan

Semua kriteria data masukan dapat dilihat pada Tabel 5.3. Tabel 5.4 menunjukkan semua kombinasi yang akan dicoba dalam proses uji coba dimana kolom Ke-1 sampai kolom Ke-5 berarti data masukan pertama sampai ke lima. Untuk setiap kasusnya, akan dilakukan dua kali uji coba dimana yang pertama dilakukan dengan memperhatikan hari libur dan yang kedua tidak.

Pemilihan data masukan didasarkan pada asumsi bahwa jumlah kendaraan yang melintas pada suatu jalan tertentu disuatu waktu tertentu akan mempunyai pola yang mirip dengan pola jumlah kendaraan yang melewati jalan tersebut dibeberapa waktu sebelumnya, misalnya dijam dan yang sama dengan waktu ramalan dihari sebelumnya. Kombinasi data masukan yang terbaik akan didapat dengan melakukan proses uji coba.

Jika hari libur ikut diperhatikan maka seperti yang telah dijelaskan, pada data masukan akan diberikan nilai 1 jika hari yang akan diramalkan adalah hari libur dan 0 jika hari yang diramalkan adalah hari aktif. Jika hari libur tidak diperhatikan maka untuk sembarang hari data masukan keenam ini akan berisi nilai 0.

Tabel 5.3 Kriteria Data Masukan

Kode	Keterangan
t-1	15 menit sebelum waktu sebenarnya
t-2	30 menit sebelum waktu sebenarnya
t-3	45 menit sebelum waktu sebenarnya
t-4	60 menit sebelum waktu sebenarnya
t-5	75 menit sebelum waktu sebenarnya
t-96	Dijam yang sama 1 hari sebelumnya
t-97	15 menit sebelum waktu sebenarnya di hari sebelumnya
t-(96×7)	Dijam dan hari yang sama 1 minggu sebelumnya
t-(96×14)	Dijam dan hari yang sama 2 minggu sebelumnya
t-(96×21)	Dijam dan hari yang sama 3 minggu sebelumnya
t-(96×28)	Dijam dan hari yang sama 4 minggu sebelumnya

Tabel 5.4 Kombinasi Data Masukan

Kasus	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5
1	t-1	t-2	t-3	t-4	t-5
2	t-1	t-(96×7)	t-(96×14)	t-(96×21)	t-(96×28)
3	t-1	t-2	t-3	t-96	t-97
4	t-96	t-(96×7)	t-(96×14)	t-(96×21)	t-(96×28)
5	t-1	t-2	t-96	t-97	t-(96×7)
6	t-1	t-2	t-3	t-(96×7)	t-(96×14)
7	t-1	t-2	t-96	t-(96×7)	t-(96×14)
8	t-1	t-2	t-3	t-4	t-96
9	t-2	t-(96×7)	t-(96×14)	t-(96×21)	t-(96×28)
10	t-4	t-(96×7)	t-(96×14)	t-(96×21)	t-(96×28)

5.4.2 Skenario 1 : Uji Coba Data Masukan Jaringan Saraf Multi Layer Perceptron

Cara perhitungan nilai kesalahan untuk metode Jaringan Saraf Tiruan Multi Layer Perceptron, yaitu pada skenario 1, 2, 3 dan 6 akan dihitung dengan cara membagi hasil peramalan yang salah dengan seluruhnya seperti pada Persamaan 5.1. x adalah jumlah peramalan yang salah dan n adalah total seluruh data.

$$Error = \frac{x}{n} \quad (5.1)$$

Hasil percobaan untuk skenario 1 ditampilkan pada Tabel 5.5. Dari hasil percobaan tersebut dapat dilihat bahwa pada banyak kasus, hasil percobaan dengan tidak memperhatikan hari libur menghasilkan hasil yang lebih baik. Perobaan dengan memperhatikan hari libur hanya menghasilkan hasil yang lebih baik pada kasus 2 dan 10. Kedua hasil uji coba tersebut menunjukkan hasil yang terbaik pada kasus kedua, yaitu dengan kombinasi data masukan t-1, t-(96×7), t-(96×14), t-(96×21) dan t-(96×28).

Tabel 5.5 Nilai Kesalahan pada Percobaan Data Masukan Jaringan Saraf Tiruan Multi Layer Perceptron

Kasus	Pengaruh Hari Libur	
	Ya	Tidak
1	20,466%	20,457%
2	11,649%	11,651%
3	17,142%	16,878%
4	11,869%	11,866%
5	13,119%	13,041%
6	12,509%	12,472%
7	12,509%	12,488%
8	19,391%	19,011%
9	11,797%	11,755%
10	12,550%	12,556%

5.4.3 Skenario 2: Uji Coba Lapisan Tersembunyi Jaringan Saraf Multi Layer Perceptron

Setelah mendapatkan kombinasi terbaik pada lapisan masukan maka untuk metode Jaringan Saraf Tiruan Multi Layer Perceptron dilakukan uji coba yang bertujuan mencari jumlah *neuron* yang terbaik (skenario 2).

Kombinasi data masukan yang dipakai adalah kombinasi data masukan kedua, karena pada percobaan sebelumnya kombinasi ini menghasilkan hasil paling maksimal Pada Tabel 5.6 dapat dilihat bahwa jumlah *neuron* pada lapisan tersembunyi yang terbaik pada kedua kasus berbeda, yaitu adalah 6. Sampai pada uji coba ini, hasil lebih baik diperlihatkan pada uji coba yang tidak memperhatikan hari libur. Pada uji coba ini dapat dilihat bahwa hasil yang didapatkan tidak terlalu fluktuasi dengan nilai kesalahan terbesar adalah 11,735% dan nilai kesalahan terkecil adalah 11,620% atau dengan kata lain antara nilai terbesar dan terkecil hanya berbeda 0,115%.

Tabel 5.6 Nilai Kesalahan pada Percobaan Jumlah Neuron pada Lapisan Tersembunyi

Jumlah Neuron	Pengaruh Hari Libur	
	Ya	Tidak
5	11,632%	11,626%
6	11,629%	11,620%
7	11,630%	11,645%
8	11,648%	11,651%
9	11,660%	11,648%
10	11,676%	11,685%
11	11,701%	11,704%
12	11,703%	11,723%
13	11,726%	11,720%
14	11,735%	11,716%

5.4.4 Skenario 3 : Uji Coba Laju Pembelajaran Pada Jaringan Saraf Tiruan Multi Layer Perceptron

Pada metode Jaringan Saraf Tiruan Multi Layer Perceptron, uji coba laju pembelajaran (skenario 3) dilakukan dengan melihat hasil percobaan sebelumnya. Pada kedua percobaan, jumlah *neuron* yang paling optimal adalah 6, kombinasi data masukan yang terbaik adalah kasus 2 dan laju pembelajaran yang membuat hasil percobaan menjadi paling baik adalah 0,0005. Hasil akhir yang didapatkan menunjukkan percobaan dengan memperhatikan hari libur mendapatkan hasil yang lebih baik walaupun hanya berbeda sedikit.

Hasil dari uji coba ini menunjukkan bahwa semakin kecil laju pembelajaran(kecuali pada nilai 0,0001) menunjukkan hasil yang semakin baik, baik itu pada percobaan dengan pengaruh hari libur maupun tanpa pengaruh hari libur. Dalam uji coba,

Tabel 5.7 Nilai Kesalahan pada Percobaan Jaringan Saraf Tiruan Multi Layer Perceptron pada Laju Pembelajaran

Laju Pembelajaran	Pengaruh Hari Libur	
	Ya (Neuron 6)	Tidak (Neuron 6)
0,0001	14,126%	14,157%
0,0005	11,502%	11,530%
0,0010	11,533%	11,533%
0,0050	11,580%	11,580%
0,0100	11,629%	11,620%
0,0500	12,062%	12,024%
0,1000	12,180%	12,189%
0,1500	12,345%	12,366%
0,2000	12,469%	12,472%
0,2500	12,562%	12,500%
0,5000	12,805%	12,780%
0,5500	12,845%	12,808%

5.4.5 Skenario 4: Uji Coba Data Masukan Jaringan Saraf Tiruan Regresi Linear

Untuk skenario 4,5 dan 6 tingkat kesalahan dihitung dengan menggunakan *Mean Absolute Percentage Error* dengan menggunakan Persamaan 5.2. n adalah total seluruh jumlah data yang diramalkan, A_t adalah hasil sebenarnya dan F_t adalah hasil peramalan.

$$Error = \frac{100\%}{n} \sum_{t=1}^n \frac{|A_t - F_t|}{A_t} \quad (5.2)$$

Hasil percobaan untuk skenario 4 dengan menggunakan metode Jaringan Saraf Tiruan Regresi Linear dapat dilihat pada Tabel 5.8. Dari hasil percobaan tersebut dapat dilihat bahwa dengan memperhatikan hari libur, hasil lebih baik didapatkan pada kasus 2, 7 dan 9. Kedua hasil percobaan tersebut mencapai hasil paling optimal dengan kombinasi data masukan kesepuluh, yaitu t-4, t-(96×7), t-(96×14), t-(96×21) dan t-(96×28).

Tabel 5.8 Nilai Kesalahan pada Percobaan Data Masukan Jaringan Saraf Tiruan Regresi Linear

Kasus	Pengaruh hari libur	
	Ya	Tidak
1	13,991%	13,900%
2	7,034%	7,047%
3	11,675%	11,503%
4	7,025%	6,985%
5	8,889%	8,844%
6	7,748%	7,733%
7	7,879%	7,844%
8	11,350%	11,164%
9	7,027%	7,049%
10	6,933%	6,925%

5.4.6 Skenario 5 : Uji Coba Laju Pembelajaran Pada Jaringan Saraf Tiruan Regresi Linear

Untuk metode Jaringan Saraf Tiruan Regresi Linear, uji coba pada laju pembelajaran menunjukan hasil yang tidak signifikan seperti yang dapat dilihat pada Tabel 5.9.

Tabel 5.9 Nilai Kesalahan pada Percobaan Jaringan Saraf Tiruan Regresi Linear pada Laju Pembelajaran

Laju Pembelajaran	Pengaruh hari libur	
	Ya (Kasus 10)	Tidak (Kasus 10)
0,0001	6,922%	6,922%
0,0005	6,923%	6,928%
0,0010	6,925%	6,925%
0,0050	6,922%	6,923%
0,0100	6,925%	6,929%
0,0500	6,924%	6,922%
0,1000	6,933%	6,925%
0,1500	6,924%	6,921%

5.4.7 Skenario 6 : Uji Coba Pada Data Pelatihan

Pada tahap ini akan dilakukan uji coba untuk kedua metode dengan menggunakan data uji coba yang lain. Inisiasi bobot pada tahap ini menggunakan bobot terakhir dari hasil uji coba. Misalnya, pada metode Jaringan Saraf Tiruan Regresi Linear dengan memperhatikan hari libur, kombinasi yang terbaik adalah dengan kombinasi data masukan ke-10 dan laju pembelajaran 0,5, maka bobot terakhir dari uji coba ini akan digunakan sebagai inisiasi bobot pada tahap uji coba. Pada tahap uji coba hanya akan dilakukan 1 kali iterasi tanpa perbaikan bobot.

Tabel 5.10 Nilai Kesalahan pada Percobaan pada Data Uji

Metode	Pengaruh hari libur	
	Ya	Tidak
Jaringan Saraf Tiruan Multi Layer Perceptron	11,739%	11,729%
Jaringan Saraf Tiruan Regresi Linear	6,934%	6,933%

5.5 Analisis

- Pada uji coba lapisan masukan untuk kedua Jaringan Saraf Tiruan, hasil yang didapatkan sangat fluktuasi, terjadi perbedaan yang cukup signifikan antara nilai kesalahan terbaik dan nilai kesalahan terburuk.
- Pada uji coba jumlah *neuron* di lapisan tersembunyi untuk Jaringan Saraf Tiruan Multi Layer Perceptron, hasil percobaan tidak menunjukkan adanya perbedaan yang cukup signifikan antara nilai kesalahan yang terbaik dan terburuk.
- Untuk kedua Jaringan Saraf Tiruan, secara umum semakin kecil nilai laju pembelajaran memberikan hasil yang semakin baik namun dengan batasan tertentu.

BAB VI

KESIMPULAN DAN SARAN

Dalam bab terakhir ini dijelaskan kesimpulan yang didapat dari pengerjaan Tugas Akhir beserta saran-saran yang dapat dipertimbangkan untuk pengembangan atau penelitian lebih lanjut.

6.1 Kesimpulan

Berdasarkan hasil uji coba yang telah dilakukan, simpulan yang bisa diambil adalah

Kedua metode Jaringan Saraf Tiruan yang digunakan dapat meramalkan jumlah kendaraan dengan baik, baik secara kategorikal maupun numerik namun membutuhkan parameter-parameter yang sesuai, baik untuk masukan pada *neuron* di lapisan masukan, jumlah *neuron* pada lapisan tersembunyi dan nilai untuk laju pembelajaran.

6.2 Saran

Adapun saran yang dapat dipertimbangkan untuk pengembangan atau penelitian lebih lanjut adalah sebagai berikut :

1. Data uji coba yang digunakan bukanlah data yang merupakan data pembangkitan namun data yang benar-benar diambil dari jalan raya.
2. Menambahkan informasi-informasi penting lainnya kedalam data masukan seperti kondisi cuaca, yang dimana sampai sekarang tidak bisa didapatkan.

DAFTAR PUSTAKA

- [1] Victorian Auditor General, Managing Traffic Congestion, 2013.
- [2] Vedat Topuz (2010). Hourly Traffic Flow Prediction Using Different ANN Models, Urban Transport and Hybrid Vehicles, Seref Soylu (Ed.), ISBN: 978-953-307-100-8, InTech,.
- [3] J. Westra, C. Bartels and P. Groeneveld, "Probabilistic Congestion Prediction," *ACM*, pp. 204-209, 2004.
- [4] J. Banks, DISCRETE EVENT SIMULATION, Engineering and Management Press, 1999.
- [5] M. Dra. Tatik Widiharhi, Buku Ajar Statistika Matematika II.
- [6] B. Baker, "Classification : Advanced Methods," in *Neural Networks(Computer Science)*, New Jersey, 1994, pp. 398-408.
- [7] Joint Research Centre (JRC), "How Many Hidden Layers and Nodes?," *International Journal of Remote Sensing*, vol. 30, pp. 2133-2147, 2009.
- [8] R. Rojas, "Statistic and Neural Networks," in *Neural Network*, Berlin, Springer-Verlag, 1996, pp. 229-263.

LAMPIRAN

Bagian ini merupakan lampiran sebagai dokumen pelengkap dari buku Tugas Akhir. Pada Lampiran A akan disajikan semua data yang telah diambil beserta cara pemodelan masukannya. Ada dua kategori data yang diambil, yaitu kategori A dimana yang dicatat adalah jarak kedatangan antara kendaraan dan kategori B menampilkan jumlah kendaraan setiap menitnya. Detail tentang waktu pengambilan data dapat dilihat pada Tabel 7.9.

Data jumlah kendaraan tersebut akan dimodelkan kedalam tiga jenis distribusi yaitu distribusi normal, *uniform* dan eksponensial. Data pada Tabel 7.1, Tabel 7.2 dan Tabel 7.3 dimodelkan kedalam distribusi eksponensial. Data pada Tabel 7.4, Tabel 7.5 dan Tabel 7.6 dimodelkan kedalam distribusi normal dan Tabel 7.7 serta Tabel 7.8 dimodelkan kedalam distribusi *uniform*. Penjelasan detail dari cara pemodelan masing-masing distribusi ini dapat dilihat pada subbab 2.1.

Tabel L1 Contoh data kategori A (1)

i	Data	x_i	D^-	D^+
1	1	0,003333	0,008161	0,003333
2	5	0,016667	0,006322	0,005172
3	7	0,023333	0,011149	0,000345
4	8	0,026667	0,01931	-0,00782
5	9	0,03	0,027471	-0,01598
6	13	0,043333	0,025632	-0,01414
7	15	0,05	0,03046	-0,01897
8	16	0,053333	0,038621	-0,02713
9	28	0,093333	0,010115	0,001379
10	29	0,096667	0,018276	-0,00678
11	34	0,113333	0,013103	-0,00161
12	40	0,133333	0,004598	0,006897
13	43	0,143333	0,006092	0,005402

i	Data	x_i	D^-	D^+
14	50	0,166667	-0,00575	0,017241
15	52	0,173333	-0,00092	0,012414
16	55	0,183333	0,000575	0,01092
17	56	0,186667	0,008736	0,002759
18	57	0,19	0,016897	-0,0054
19	63	0,21	0,008391	0,003103
20	67	0,223333	0,006552	0,004943
21	68	0,226667	0,014713	-0,00322
22	69	0,23	0,022874	-0,01138
23	70	0,233333	0,031034	-0,01954
24	73	0,243333	0,032529	-0,02103
25	78	0,26	0,027356	-0,01586
26	85	0,283333	0,015517	-0,00402
27	89	0,296667	0,013678	-0,00218
28	90	0,3	0,021839	-0,01034
29	93	0,31	0,023333	-0,01184
30	98	0,326667	0,018161	-0,00667
31	103	0,343333	0,012989	-0,00149
32	113	0,376667	-0,00885	0,020345
33	116	0,386667	-0,00736	0,018851
34	118	0,393333	-0,00253	0,014023
35	119	0,396667	0,005632	0,005862
36	122	0,406667	0,007126	0,004368
37	126	0,42	0,005287	0,006207
38	131	0,436667	0,000115	0,011379
39	133	0,443333	0,004943	0,006552
40	136	0,453333	0,006437	0,005057
41	138	0,46	0,011264	0,00023
42	141	0,47	0,012759	-0,00126
43	145	0,483333	0,01092	0,000575
44	146	0,486667	0,01908	-0,00759
45	148	0,493333	0,023908	-0,01241
46	150	0,5	0,028736	-0,01724
47	155	0,516667	0,023563	-0,01207
48	160	0,533333	0,018391	-0,0069
49	161	0,536667	0,026552	-0,01506

i	Data	x_i	D^-	D^+
50	164	0,546667	0,028046	-0,01655
51	166	0,553333	0,032874	-0,02138
52	167	0,556667	0,041034	-0,02954
53	172	0,573333	0,035862	-0,02437
54	175	0,583333	0,037356	-0,02586
55	181	0,603333	0,028851	-0,01736
56	186	0,62	0,023678	-0,01218
57	192	0,64	0,015172	-0,00368
58	198	0,66	0,006667	0,004828
59	202	0,673333	0,004828	0,006667
60	203	0,676667	0,012989	-0,00149
61	215	0,716667	-0,01552	0,027011
62	215	0,716667	-0,00402	0,015517
63	216	0,72	0,004138	0,007356
64	223	0,743333	-0,0077	0,019195
65	225	0,75	-0,00287	0,014368
66	231	0,77	-0,01138	0,022874
67	233	0,776667	-0,00655	0,018046
68	236	0,786667	-0,00506	0,016552
69	238	0,793333	-0,00023	0,011724
70	242	0,806667	-0,00207	0,013563
71	243	0,81	0,006092	0,005402
72	244	0,813333	0,014253	-0,00276
73	245	0,816667	0,022414	-0,01092
74	247	0,823333	0,027241	-0,01575
75	248	0,826667	0,035402	-0,02391
76	253	0,843333	0,03023	-0,01874
77	261	0,87	0,015057	-0,00356
78	262	0,873333	0,023218	-0,01172
79	263	0,876667	0,031379	-0,01989
80	266	0,886667	0,032874	-0,02138
81	269	0,896667	0,034368	-0,02287
82	274	0,913333	0,029195	-0,0177
83	276	0,92	0,034023	-0,02253
84	281	0,936667	0,028851	-0,01736
85	285	0,95	0,027011	-0,01552

i	Data	x_i	D^-	D^+
86	293	0,976667	0,011839	-0,00034
87	295	0,983333	0,016667	-0,00517

Nilai hitung 0,1438 dan nilai tabel 0,145 sehingga data tersebut terdistribusi secara eksponensial.

Tabel L2 Contoh data kategori A (2)

i	Data	x_i	D^-	D^+
1	4	0,013333	0,002051	0,013333
2	7	0,023333	0,007436	0,007949
3	8	0,026667	0,019487	-0,0041
4	13	0,043333	0,018205	-0,00282
5	14	0,046667	0,030256	-0,01487
6	23	0,076667	0,015641	-0,00026
7	26	0,086667	0,021026	-0,00564
8	42	0,14	-0,01692	0,032308
9	49	0,163333	-0,02487	0,040256
10	52	0,173333	-0,01949	0,034872
11	57	0,19	-0,02077	0,036154
12	61	0,203333	-0,01872	0,034103
13	61	0,203333	-0,00333	0,018718
14	62	0,206667	0,008718	0,006667
15	68	0,226667	0,004103	0,011282
16	78	0,26	-0,01385	0,029231
17	91	0,303333	-0,04179	0,057179
18	91	0,303333	-0,02641	0,041795
19	94	0,313333	-0,02103	0,03641
20	95	0,316667	-0,00897	0,024359
21	96	0,32	0,003077	0,012308
22	97	0,323333	0,015128	0,000256
23	100	0,333333	0,020513	-0,00513
24	109	0,363333	0,005897	0,009487
25	111	0,37	0,014615	0,000769
26	116	0,386667	0,013333	0,002051
27	121	0,403333	0,012051	0,003333
28	132	0,44	-0,00923	0,024615

i	Data	x_i	D^-	D^+
29	133	0,443333	0,002821	0,012564
30	138	0,46	0,001538	0,013846
31	146	0,486667	-0,00974	0,025128
32	148	0,493333	-0,00103	0,01641
33	149	0,496667	0,011026	0,004359
34	150	0,5	0,023077	-0,00769
35	150	0,5	0,038462	-0,02308
36	150	0,5	0,053846	-0,03846
37	154	0,513333	0,055897	-0,04051
38	155	0,516667	0,067949	-0,05256
39	174	0,58	0,02	-0,00462
40	178	0,593333	0,022051	-0,00667
41	178	0,593333	0,037436	-0,02205
42	187	0,623333	0,022821	-0,00744
43	188	0,626667	0,034872	-0,01949
44	191	0,636667	0,040256	-0,02487
45	193	0,643333	0,048974	-0,03359
46	202	0,673333	0,034359	-0,01897
47	202	0,673333	0,049744	-0,03436
48	205	0,683333	0,055128	-0,03974
49	205	0,683333	0,070513	-0,05513
50	210	0,7	0,069231	-0,05385
51	219	0,73	0,054615	-0,03923
52	232	0,773333	0,026667	-0,01128
53	235	0,783333	0,032051	-0,01667
54	239	0,796667	0,034103	-0,01872
55	242	0,806667	0,039487	-0,0241
56	253	0,843333	0,018205	-0,00282
57	255	0,85	0,026923	-0,01154
58	257	0,856667	0,035641	-0,02026
59	262	0,873333	0,034359	-0,01897
60	262	0,873333	0,049744	-0,03436
61	264	0,88	0,058462	-0,04308
62	265	0,883333	0,070513	-0,05513
63	272	0,906667	0,062564	-0,04718
64	277	0,923333	0,061282	-0,0459

Nilai hitung 0,156 dan nilai tabel 0,168 sehingga data tersebut terdistribusi secara eksponensial.

Tabel L3 Contoh data kategori A (3)

i	Data	x_i	D^-	D^+
1	2	0,006667	0,017724	0,006667
2	3	0,01	0,03878	-0,01439
3	7	0,023333	0,049837	-0,02545
4	10	0,033333	0,064228	-0,03984
5	13	0,043333	0,078618	-0,05423
6	19	0,063333	0,083008	-0,05862
7	24	0,08	0,090732	-0,06634
8	35	0,116667	0,078455	-0,05407
9	37	0,123333	0,096179	-0,07179
10	49	0,163333	0,080569	-0,05618
11	71	0,236667	0,031626	-0,00724
12	77	0,256667	0,036016	-0,01163
13	82	0,273333	0,04374	-0,01935
14	87	0,29	0,051463	-0,02707
15	100	0,333333	0,03252	-0,00813
16	100	0,333333	0,056911	-0,03252
17	106	0,353333	0,061301	-0,03691
18	110	0,366667	0,072358	-0,04797
19	120	0,4	0,063415	-0,03902
20	129	0,43	0,057805	-0,03341
21	137	0,456667	0,055528	-0,03114
22	138	0,46	0,076585	-0,0522
23	139	0,463333	0,097642	-0,07325
24	150	0,5	0,085366	-0,06098
25	162	0,54	0,069756	-0,04537
26	172	0,573333	0,060813	-0,03642
27	190	0,633333	0,025203	-0,00081
28	190	0,633333	0,049593	-0,0252
29	216	0,72	-0,01268	0,037073
30	217	0,723333	0,008374	0,016016
31	217	0,723333	0,032764	-0,00837

i	Data	x_i	D^-	D^+
32	218	0,726667	0,053821	-0,02943
33	231	0,77	0,034878	-0,01049
34	238	0,793333	0,035935	-0,01154
35	244	0,813333	0,040325	-0,01593
36	255	0,85	0,028049	-0,00366
37	259	0,863333	0,039106	-0,01472
38	262	0,873333	0,053496	-0,02911
39	295	0,983333	-0,03211	0,056504
40	298	0,993333	-0,01772	0,042114
41	299	0,996667	0,003333	0,021057

Nilai hitung 0,201 dan nilai tabel 0,212 sehingga data tersebut terdistribusi secara eksponensial.

Tabel L4 Contoh data kategori B (1)

i	x_i	$F(x_i)$	i/n	$\frac{i}{n} - F(x_i)$	$F(x_i) - (\frac{i-1}{n})$
1	7	0,333333	0,033333	-0,3	0,333333
2	7	0,333333	0,066667	-0,26667	0,3
3	8	0,380952	0,1	-0,28095	0,314286
4	9	0,428571	0,133333	-0,29524	0,328571
5	9	0,428571	0,166667	-0,2619	0,295238
6	10	0,47619	0,2	-0,27619	0,309524
7	10	0,47619	0,233333	-0,24286	0,27619
8	10	0,47619	0,266667	-0,20952	0,242857
9	10	0,47619	0,3	-0,17619	0,209524
10	11	0,52381	0,333333	-0,19048	0,22381
11	12	0,571429	0,366667	-0,20476	0,238095
12	12	0,571429	0,4	-0,17143	0,204762
13	12	0,571429	0,433333	-0,1381	0,171429
14	12	0,571429	0,466667	-0,10476	0,138095
15	12	0,571429	0,5	-0,07143	0,104762
16	12	0,571429	0,533333	-0,0381	0,071429
17	13	0,619048	0,566667	-0,05238	0,085714

i	x_i	$F(x_i)$	i/n	$\frac{i}{n} - F(x_i)$	$F(x_i) - (\frac{i-1}{n})$
18	13	0,619048	0,6	-0,01905	0,052381
19	13	0,619048	0,633333	0,014286	0,019048
20	13	0,619048	0,666667	0,047619	-0,01429
21	13	0,619048	0,7	0,080952	-0,04762
22	14	0,666667	0,733333	0,066667	-0,03333
23	14	0,666667	0,766666	0,1	-0,06667
24	14	0,666667	0,8	0,133333	-0,1
25	17	0,809524	0,833333	0,02381	0,009524
26	17	0,809524	0,866666	0,057143	-0,02381
27	18	0,857143	0,9	0,042857	-0,00952
28	18	0,857143	0,933333	0,07619	-0,04286
29	20	0,952381	0,966666	0,014286	0,019048
30	21	1	1	0	0,033333

Nilai hitung -0,264 dan nilai tabel 0,242 sehingga data tersebut terdistribusi secara normal.

Tabel L5 Contoh data kategori B (2)

i	x_i	$F(x_i)$	i/n	$\frac{i}{n} - F(x_i)$	$F(x_i) - (\frac{i-1}{n})$
1	5	0,357143	0,033333	-0,32381	0,357143
2	8	0,571429	0,066667	-0,50476	0,538095
3	8	0,571429	0,1	-0,47143	0,504762
4	8	0,571429	0,133333	-0,4381	0,471429
5	9	0,642857	0,166667	-0,47619	0,509524
6	9	0,642857	0,2	-0,44286	0,47619
7	9	0,642857	0,233333	-0,40952	0,442857
8	9	0,642857	0,266667	-0,37619	0,409524
9	9	0,642857	0,3	-0,34286	0,37619
10	10	0,714286	0,333333	-0,38095	0,414286
11	10	0,714286	0,366667	-0,34762	0,380952
12	10	0,714286	0,4	-0,31429	0,347619
13	10	0,714286	0,433333	-0,28095	0,314286

i	x_i	$F(x_i)$	i/n	$\frac{i}{n} - F(x_i)$	$F(x_i) - (\frac{i-1}{n})$
14	10	0,714286	0,466667	-0,24762	0,280952
15	10	0,714286	0,5	-0,21429	0,247619
16	11	0,785714	0,533333	-0,25238	0,285714
17	11	0,785714	0,566667	-0,21905	0,252381
18	11	0,785714	0,6	-0,18571	0,219048
19	11	0,785714	0,633333	-0,15238	0,185714
20	11	0,785714	0,666667	-0,11905	0,152381
21	11	0,785714	0,7	-0,08571	0,119048
22	11	0,785714	0,733333	-0,05238	0,085714
23	12	0,857143	0,766667	-0,09048	0,12381
24	12	0,857143	0,8	-0,05714	0,090476
25	12	0,857143	0,833333	-0,02381	0,057143
26	12	0,857143	0,866667	0,009524	0,02381
27	12	0,857143	0,9	0,042857	-0,00952
28	12	0,857143	0,933333	0,07619	-0,04286
29	13	0,928571	0,966667	0,038095	-0,00476
30	14	1	1	0	0,033333

Nilai hitung -0,221 dan nilai tabel 0,242 sehingga data tersebut terdistribusi secara normal.

Tabel L6 Contoh data kategori B (3)

i	x_i	$F(x_i)$	i/n	$\frac{i}{n} - F(x_i)$	$F(x_i) - (\frac{i-1}{n})$
1	9	0,529412	0,033333	-0,49608	0,529412
2	9	0,529412	0,066667	-0,46275	0,496078
3	9	0,529412	0,1	-0,42941	0,462745
4	10	0,588235	0,133333	-0,4549	0,488235
5	11	0,647059	0,166667	-0,48039	0,513725
6	11	0,647059	0,2	-0,44706	0,480392
7	11	0,647059	0,233333	-0,41373	0,447059
8	11	0,647059	0,266667	-0,38039	0,413725
9	12	0,705882	0,3	-0,40588	0,439216

i	x_i	$F(x_i)$	i/n	$\frac{i}{n} - F(x_i)$	$F(x_i) - (\frac{i-1}{n})$
10	12	0,705882	0,333333	-0,37255	0,405882
11	12	0,705882	0,366667	-0,33922	0,372549
12	12	0,705882	0,4	-0,30588	0,339216
13	13	0,764706	0,433333	-0,33137	0,364706
14	13	0,764706	0,466667	-0,29804	0,331373
15	13	0,764706	0,5	-0,26471	0,298039
16	13	0,764706	0,533333	-0,23137	0,264706
17	14	0,823529	0,566667	-0,25686	0,290196
18	14	0,823529	0,6	-0,22353	0,256863
19	14	0,823529	0,633333	-0,1902	0,223529
20	14	0,823529	0,666667	-0,15686	0,190196
21	15	0,882353	0,7	-0,18235	0,215686
22	15	0,882353	0,733333	-0,14902	0,182353
23	15	0,882353	0,766667	-0,11569	0,14902
24	15	0,882353	0,8	-0,08235	0,115686
25	15	0,882353	0,833333	-0,04902	0,082353
26	16	0,941176	0,866667	-0,07451	0,107843
27	16	0,941176	0,9	-0,04118	0,07451
28	16	0,941176	0,933333	-0,00784	0,041176
29	16	0,941176	0,966667	0,02549	0,007843
30	17	1	1	0	0,033333

Nilai hitung -0,126 dan nilai tabel 0,242 sehingga data tersebut terdistribusi secara normal.

Tabel L7 Contoh data kategori B (4)

x_i	Frekuensi	Frekuensi Relatif	$Fa(x_i)$	$Fe(x_i)$	$ Fa(x_i) - Fe(x_i) $
4	2	0,066667	0,066667	0,222222	0,155556
5	2	0,066667	0,133333	0,277778	0,144444
6	3	0,1	0,233333	0,333333	0,1
7	2	0,066667	0,3	0,388889	0,088889
8	3	0,1	0,4	0,444444	0,044444

x_i	Frekuensi	Frekuensi Relatif	$Fa(x_i)$	$Fe(x_i)$	$ Fa(x_i) - Fe(x_i) $
10	2	0,066667	0,533333	0,555556	0,022222
11	2	0,066667	0,6	0,611111	0,011111
12	3	0,1	0,7	0,666667	0,033333
13	2	0,066667	0,766667	0,722222	0,044444
14	3	0,1	0,866667	0,777778	0,088889
16	2	0,066667	0,933333	0,888889	0,044444
18	2	0,066667	1	1	0

Nilai hitung 0,144 dan nilai tabel 0,242 sehingga data tersebut terdistribusi secara uniform.

Tabel L8 Contoh data kategori B (5)

x_i	Frekuensi	Frekuensi Relatif	$Fa(x_i)$	$Fe(x_i)$	$ Fa(x_i) - Fe(x_i) $
10	8	0,266667	0,266667	0,5	0,233333
11	4	0,133333	0,4	0,55	0,15
12	4	0,133333	0,533333	0,6	0,066667
13	2	0,066667	0,6	0,65	0,05
14	3	0,1	0,7	0,7	0
15	5	0,166667	0,866667	0,75	0,116667
16	4	0,133333	1	0,8	0,2

Nilai hitung 0,233 dan nilai tabel 0,242 sehingga data tersebut terdistribusi secara uniform.

Tabel L9 Detail waktu pengambilan data

Tabel	Waktu
Tabel 7.1	4 Juni 2014, antara jam 10.00-11.00
Tabel 7.2	5 Juni 2014, antara jam 18.00-19.00
Tabel 7.3	7 Juni 2014, antara jam 18.00-19.00
Tabel 7.4	5 Juni 2014, antara jam 08.00-09.00
Tabel 7.5	5 Juni 2014, antara jam 14.00-15.00
Tabel 7.6	7 Juni 2014, antara jam 08.00-09.00
Tabel 7.7	8 Juni 2014, antara jam 10.00-11.00
Tabel 7.8	8 Juni 2014, antara jam 14.00-15.00

BIODATA PENULIS



Gregorius Edwadr, dilahirkan di desa Reo, Nusa Tenggara Timur pada tanggal 11 Juli 1992. Penulis adalah anak kedua dari dua bersaudara. Penulis telah menempuh pendidikan formal yaitu di SDK Reo III (1998-2004), SMP YPPI 1 Surabaya (2004-2007) dan SMAK Frateran Surabaya (2007-2010). Setelah lulus dari SMAK Frateran Surabaya pada tahun 2010, Penulis mengikuti SNMPTN dan diterima di Jurusan Teknik Informatika ITS pada tahun 2010 dan terdaftar dengan NRP 5110100188. Selama menjadi mahasiswa, penulis pernah berperan sebagai asisten praktikum Mata Kuliah Pemrograman Terstruktur serta Algoritma dan Struktur Data, dan asisten dosen Mata Kuliah Algoritma dan Struktur Data. Di Jurusan Teknik Informatika ITS ini, Penulis mengambil Bidang Minat Komputasi Cerdas dan Visualisasi (KCV). Penulis dapat dihubungi melalui alamat *e-mail* di sh_gedward@live.com.